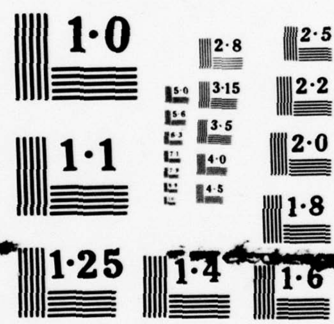
AD-A055 946 DELAWARE UNIV NEWARK DEPT OF STATISTICS AND COMPUTER--ETC F/G 12/1
NUMERICAL METHODS FOR A CLASS OF MARKOV CHAINS ARISING IN QUEUE--ETC(U)
MAY 78 D M LUCANTONI, M F NEUTS AFOSR-77-3236
UNCLASSIFIED TR-78/10 AFOSR-TR-78-1049 NL

1 OF 2
ADA
055946



AFOSR-TR- 78-1049

FOR FURTHER TRAN

2

AD A055946

AD No.
DDC FILE COPY

Department of

STATISTICS AND COMPUTER SCIENCE

D D C
JUN 21 1978
E

UNIVERSITY OF DELAWARE
Newark, Delaware 19711

78 06 19

(2)

NUMERICAL METHODS FOR A CLASS OF MARKOV CHAINS ARISING IN QUEUEING THEORY

by

David Michael Lucantoni

and

Marcel F. Neuts

Department of Statistics
and
Computer Science
Technical Report No. 78/10

May 1978

This research was sponsored by the Air Force Office of Scientific Research Air Force Systems Command USAF, under Grant No. AFOSR-77-3236. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 18 AFOSR TR-78-1049	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 14 TR-78/19
4. TITLE (and Subtitle) 6 NUMERICAL METHODS FOR A CLASS OF MARKOV CHAINS ARISING IN QUEUEING THEORY.		5. TYPE OF REPORT & PERIOD COVERED 9 Interim rept.,
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 David Michael Lucantoni ~~and~~ Marcel F. Neuts		8. CONTRACT OR GRANT NUMBER(s) 15 AFOSR-77-3236
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Delaware Department of Statistics & Computer Science Newark, Delaware 19711		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 16 2304 17 A5
11. CONTROLLING OFFICE NAME AND ADDRESS Air For Office of Scientific Research/NM Bolling AFB, Washington, DC 20332		12. REPORT DATE 11 May 1978
		13. NUMBER OF PAGES 99 12 103p.
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computational probability, Queueing theory, Markov chains, steady-state queue length, block-partitioned stochastic matrices.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

We discuss an algorithm is discussed for computing the stationary probability vector of an infinite-state Markov chain whose transition probability matrix has a block-partitioned structure. Such matrices arise in a wide variety of queueing models as well as generalized random walk problems. Traditionally, the analytic approach to this type of problem has been through complex variable methods. ~~We present~~ an alternate and unified treatment of this problem is presented and ~~obtain~~ an algorithm is obtained which → next page

DD FORM 1 JAN 73 1473

UNCLASSIFIED

391 807

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Abstract

cont utilizes only real arithmetic computations. In addition, most of the intermediate steps of the algorithm have useful probabilistic interpretations.

We obtain an adequate number of the initial components of the invariant vector by using a purely probabilistic argument. Higher components are evaluated by matrix-iterative methods. The first and second moments of the stationary distribution are also found in computationally tractable forms. The APL program used to implement the algorithm is listed and several numerical examples are presented.

ABSTRACT

We discuss an algorithm for computing the stationary probability vector of an infinite-state Markov chain whose transition probability matrix has a block-partitioned structure. Such matrices arise in a wide variety of queueing models as well as generalized random walk problems. Traditionally, the analytic approach to this type of problem has been through complex variable methods. We present an alternate and unified treatment of this problem and obtain an algorithm which utilizes only real arithmetic computations. In addition, most of the intermediate steps of the algorithm have useful probabilistic interpretations.

We obtain an adequate number of the initial components of the invariant vector by using a purely probabilistic argument. Higher components are evaluated by matrix-iterative methods. The first and second moments of the stationary distribution are also found in computationally tractable forms. The APL program used to implement the algorithm is listed and several numerical examples are presented.

KEY WORDS

Computational probability, Queueing theory, Markov chains, steady-state queue length, block-partitioned stochastic matrices.

<u>GLOSSARY</u>

A	- the irreducible stochastic matrix $\sum_{\nu=0}^{\infty} A_\nu$.
$\{A_n\}_{n=0}^{\infty}$	- a sequence of nonnegative m×m matrices whose sum is stochastic and irreducible.
$A^*(z)$	- the matrix generating function $\sum_{\nu=0}^{\infty} A_\nu z^\nu$.
$A_\nu^*(z)$	- the matrix obtained by differentiating the matrix $A^*(z)$ entrywise, ν times with respect to z.
A_k'	- $A_k(I-A_1)^{-1}$.
$\{B_\nu\}_{\nu=0}^{\infty}$	- a sequence of nonnegative matrices satisfying $B_0\underline{e} + \sum_{\nu=1}^{\infty} B_\nu\underline{e} = \underline{e}$ where B_0 is n*n and B_k is n×m for $k \geq 1$.
$\underline{b}_k'$	- $(\underline{x}_0 B_k + \underline{x}_1 A_k)(I-A_1)^{-1}$.
$\underline{d}$	- the invariant probability vector of the matrix L.
$\underline{d}^*$	- the mean vector $\left[\frac{d}{dz}L(z)\right]_{z=1}\underline{e}$.
$\underline{e}$	- a vector (of appropriate dimension) each of whose components is 1.
E_j	- the mean recurrence time for the state (0,j).
G	- $G(z)\|_{z=1}$.

$G(k)$ - the matrix $\{G_{jj'}(k)\}$ where $G_{jj'}(k)$ equals the probability that starting in state $(i+1,j)$, $i \geq 0$, the level $\underline{i}$ is reached for the first time in state (i,j') in exactly k transitions.

$G(z)$ - the transform matrix $\sum_{k=0}^{\infty} G(k) z^k$ satisfying the functional equation $G(z) = \sum_{\nu=0}^{\infty} zA_\nu G^\nu(z)$.

$\tilde{G}$ - the matrix $\{G_{jj'}\}$ where $G_{jj'} = g_{j'}$, for $1 \leq j, j' \leq m$.

$\underline{g}$ - the stationary probability vector of G.

H - $H(z)|_{z=1}$.

$H(k)$ - the matrix $\{H_{jj'}(k)\}$, where $H_{jj'}(k)$ equals the probability that starting in state $(1,j)$, the level $\underline{0}$ is reached for the first time in state $(0,j')$ in exactly k transitions.

$H(z)$ - the transform matrix $\sum_{k=0}^{\infty} H(k) z^k$.

$\underline{h}^*$ - the mean vector $\left[\frac{d}{dz} H(z)\right]_{z=1} \underline{e}$.

I - the $m \times m$ identity matrix.

$\underline{i}$ - the "level i", where $i \geq 1$, consisting of the set of states $\{(i,j), 1 \leq j \leq m\}$ in the infinite Markov chain P.

K - $K(z)|_{z=1}$

$\mathbf{K}(k)$ - the matrix $\{K_{jj'}(k)\}$ where $K_{jj'}(k)$ equals the probability that starting in state $(1,j)$, the Markov chain returns to level $\underline{1}$ for the first time in the state $(1,j')$ in exactly k transitions.

$\mathbf{K}(z)$ - the transform matrix $\sum_{k=0}^{\infty} \mathbf{K}(k) z^k$.

$\mathbf{L}$ - $\mathbf{L}(z)|_{z=1}$.

$\mathbf{L}(k)$ - $\{L_{jj'}(k)\}$ where $L_{jj'}(k)$ equals the probability that starting in state $(0,j)$, the Markov chain returns to level $\underline{0}$ for the first time in the state $(0,j')$ in exactly k transitions.

$\mathbf{L}(z)$ - the transform matrix $\sum_{k=0}^{\infty} \mathbf{L}(k) z^k$.

P - the transition probability matrix of the infinite Markov chain having the particular structure of interest.

$\underline{X}(z)$ - the vector generating function $\sum_{\nu=1}^{\infty} \underline{x}_\nu z^\nu$.

$\underline{X}^{(n)}(z)$ - the vector obtained by differentiating the vector $\underline{X}(z)$, n times with respect to z.

$\underline{x}$ - the invariant probability vector of the matrix P.

$\underline{u}(z)$ - the appropriately normalized right eigenvector of the matrix $A^*(z)$, corresponding to the Perron-Frobenius eigenvalue.

$\underline{u}^{(n)}(z)$ - the vector obtained by differentiating the vector $\underline{u}(z)$, n times with respect to z.

$\underline{v}(z)$ - the appropriately normalized left eigenvector of the matrix $A^*(z)$, corresponding to the Perron-Frobenius eigenvalue.

$\underline{v}^{(n)}(z)$ - the vector obtained by differentiating the vector $\underline{v}(z)$, n times with respect to z.

$\underline{\beta}$ - $\sum_{\nu=0}^{\infty} \nu A_\nu \underline{e}$.

$\delta(z)$ - the Perron-Frobenius eigenvalue of the matrix $A^*(z)$.

$\delta^{(n)}(z)$ - the n-th derivative of $\delta(z)$ with respect to z.

$\Delta(\underline{x})$ - a diagonal matrix with the elements of $\underline{x}$ along the diagonal.

$\underline{\kappa}$ - the invariant probability vector of the matrix K.

$\underline{\kappa}^*$ - the mean vector $\left[\frac{d}{dz} K(z)\right]_{z=1} \underline{e}$.

$\underline{\mu}$ - the mean vector $\left[\frac{d}{dz} G(z)\right]_{z=1} \underline{e}$.

$\underline{\pi}$ - the invariant probability vector of the matrix A.

Π - the matrix $\{\Pi_{jj'}\}$ where $\Pi_{jj'} = \pi_{j'}$.

ρ - the quantity $\underline{\pi}\underline{\beta}$.

1. INTRODUCTION

We are concerned with a class of infinite Markov chains with stationary transition probabilities, having a transition matrix P of the following form:

$$(1) \qquad P = \begin{bmatrix} B_0 & B_1 & B_2 & B_3 & B_4 & \cdots \\ C_0 & A_1 & A_2 & A_3 & A_4 & \cdots \\ 0 & A_0 & A_1 & A_2 & A_3 & \cdots \\ 0 & 0 & A_0 & A_1 & A_2 & \cdots \\ 0 & 0 & 0 & A_0 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \end{bmatrix},$$

where the matrices A_ν, $\nu \geq 0$, are square substochastic matrices of order m. The matrices B_ν, $\nu \geq 1$, are n×m, while B_0 is n×n and C_0 is m×n. The state space of this Markov chain is the set $\{(0,j), 1 \leq j \leq n$ and $(i,j), i \geq 1, 1 \leq j \leq m\}$. The matrix $A = \sum_{\nu=0}^{\infty} A_\nu$ is stochastic and

$$(2) \qquad C_0\underline{e} + \sum_{\nu=1}^{\infty} A_\nu \underline{e} = \underline{e},$$

$$B_0\underline{e} + \sum_{\nu=1}^{\infty} B_\nu \underline{e} = \underline{e},$$

where $\underline{e} = (1,1,\ldots,1)'$. We shall derive an algorithm to

compute the steady-state probability vector $\underline{x}$ = $(\underline{x}_0, \underline{x}_1, \underline{x}_2, \ldots)$ of the matrix P, where $\underline{x}_0$ is an n-vector and $\underline{x}_k$, $k \geq 1$, are m-vectors. This amounts to solving the infinite system of linear equations

$$(3) \qquad \underline{x}\, P = \underline{x}, \qquad \underline{x}\, \underline{e} = 1,$$

or equivalently,

$$(4) \qquad \underline{x}_0 = \underline{x}_0 B_0 + \underline{x}_1 C_0$$

$$\underline{x}_k = \underline{x}_0 B_k + \sum_{\nu=1}^{k+1} \underline{x}_\nu A_{k+1-\nu}, \text{ for } k \geq 1.$$

We define the probability generating function vector $\underline{X}(z)$, $0 \leq z \leq 1$, as

$$(5) \qquad \underline{X}(z) = \sum_{k=1}^{\infty} \underline{x}_k z^k$$

$$= \underline{x}_0 \sum_{k=1}^{\infty} B_k z^k + \sum_{k=1}^{\infty} z^k \sum_{\nu=1}^{k+1} \underline{x}_\nu A_{k+1-\nu},$$

and the generating function $A^*(z)$, $0 \leq z \leq 1$, of the sequence of matrices $\{A_n\}$ as

$$(6) \qquad A^*(z) = \sum_{\nu=0}^{\infty} A_\nu z^\nu.$$

If we interchange the order of summation in the second term on the right of Equation (5), we have

$$\underline{X}(z) = \underline{x}_0 \sum_{k=1}^{\infty} B_k z^k + z^{-1} \sum_{\nu=1}^{\infty} \underline{x}_\nu z^\nu \sum_{k=\nu-1}^{\infty} z^{k-\nu+1} A_{k-\nu+1} - \underline{x}_1 A_0$$

$$= \underline{x}_0 \sum_{k=1}^{\infty} B_k z^k + z^{-1} \underline{X}(z) A^*(z) - \underline{x}_1 A_0,$$

and therefore

$$(7) \qquad \underline{X}(z)[zI - A^*(z)] = z\, \underline{x}_0 \sum_{k=1}^{\infty} B_k z^k - z\underline{x}_1 A_0.$$

It is traditional to attempt to derive the vector $\underline{x}_0$ from Equation (7) by using complex variable methods, based on an application of Rouché's theorem. In practice, however, this method may lead to highly unstable numerical computations. We shall derive the vector $\underline{x}_0$ using a purely probabilistic argument. It should be stressed that our approach will utilize only real-arithmetic algorithms and so avoids many of the numerical problems associated with the complex variable methodology. Our discussion reviews and generalizes a number of earlier results, used in the analysis of specific queueing models [24,25,27].

Markov chains of the type (1) appear as the embedded Markov chains in a large number of queueing models. Computing the vector $\underline{x}$ is a crucial step in the numerical evaluation of many quantities and probability distributions of relevance to the theory of queues. A list of substantially different queueing models, which are amenable to the present analysis is given in Section 8.

2. THE FIRST PASSAGE TIMES FROM LEVEL $\underline{i+1}$ TO LEVEL $\underline{i}$

Consider the first passage times from the set of states $\underline{i+1} = \{(i+1,j),\ j\varepsilon(1,\ldots,m)\}$ to the set of states $\underline{i} = \{(i,j), j\varepsilon(1,\ldots,m)\}$, $i\geq 1$. The set $\underline{i}$ will henceforth be referred to as level $\underline{i}$. Let $G_{jj'}(k), 1\leq j,j'\leq m$, $k\geq 1$, be the conditional probability that, starting in the state $(i+1,j)$, the process reaches the level $\underline{i}$ for the first time in the state (i,j') after exactly k transitions. Define the sequence of matrices $\{G(k), k\geq 1\}$ such that $G(k)=\{G_{jj'}(k)\}$. This sequence of matrices defines completely the first passage time distributions from states in the level $\underline{i+1}$ to the level $\underline{i}$. These matrices were studied in great detail by Neuts[21,24]. We now review a number of important results from these papers, which are needed in the sequel.

The matrices $G(k), k\geq 1$, are most conveniently studied by considering the matrix of transforms $G(z)$, defined by

$$(8) \qquad G(z) = \sum_{k=1}^{\infty} G(k) z^k, \quad \text{for } 0\leq z\leq 1.$$

By using a standard first passage argument, it is shown that $G(z)$ satisfies the matrix functional equation

$$(9) \qquad G(z) = \sum_{\nu=0}^{\infty} zA_\nu G^\nu(z).$$

In [21] it is shown that Equation (9) uniquely determines the sequence of matrices $\{G(k), k \geq 1\}$. For the process eventually to reach level $\underline{i}$ from any state in $\underline{i+1}$, the matrix $G=G(1-)$ must be stochastic. The following theorem is proved in [21], assuming certain irreducibility conditions, which are generally satisfied in applications and which we shall not repeat here.

Theorem 1: Let $\underline{\pi}$ be the invariant probability vector of the irreducible stochastic matrix A, i.e., the unique solution to the equations

$$(10) \qquad \underline{\pi}\, A = \underline{\pi} \quad \text{and} \quad \underline{\pi}\, \underline{e} = 1.$$

Also let $\underline{\beta} = \sum_{\nu=1}^{\infty} \nu A_\nu \underline{e}$. Then the equation

$$(11) \qquad G = \sum_{\nu=0}^{\infty} A_\nu G^\nu,$$

has a minimal nonnegative solution which is stochastic if and only if $\rho = \underline{\pi}\, \underline{\beta} \leq 1$. The matrix G is then also the unique nonnegative matrix satisfying that equation.

Remarks:

a) For what follows, we assume the above mentioned irreducibility conditions hold.

b) In the queueing context, ρ as defined above, represents the expected number of arrivals during a suitably averaged service time and corresponds to the familiar <u>traffic intensity</u>.

c) If $\rho \geq 1$, the chain is null-recurrent or transient and therefore no solution to the equations (3) exists. In the sequel, only the case $\rho < 1$ is discussed.

d) The matrix G may most conveniently be computed by modified successive substitutions. This corresponds to successively evaluating the matrices

$$\text{(12)} \qquad \hat{G}(0) = (I-A_1)^{-1}A_0,$$

$$\hat{G}(k+1) = \sum_{\substack{\nu=0 \\ \nu\neq 1}}^{\infty} (I-A_1)^{-1}A_\nu G^\nu(k), \text{ for } k \geq 0.$$

It was shown in [21], that this sequence is entry-wise strictly increasing and converges to the matrix G.

For future reference, we introduce the vector $\underline{g}$ of stationary probabilities corresponding to the stochastic matrix G and the square matrix $\tilde{G}$ of order m, whose rows are all identical and equal to the vector $\underline{g}$. Since the irreducible matrix G has spectral radius equal to one, the matrix (I-G) is singular. It is an important and well-known result though, that the matrix $(I-G+\tilde{G})$ is non-singular. (see Kemeny and Snell [11]). We shall also need the

mean vector $\underline{\mu}$ defined by

(13) $$\underline{\mu} = \sum_{k=1}^{\infty} k\, G(k)\underline{e}.$$

Theorem 2: The vector $\underline{\mu}$ is given by

(14) $$\underline{\mu} = (I-G+\tilde{G})[I-A+\tilde{G}-\Delta(\underline{\beta})\tilde{G}]^{-1}\underline{e}$$

where $\Delta(\underline{\beta})$ is a diagonal matrix of order m, with diagonal entries $\beta_1, \beta_2, \ldots, \beta_m$.

Proof:

(15) $$\underline{\mu} = \left[\frac{d}{dz} G(z)\right]_{z=1}\underline{e}$$

$$= \left[\sum_{\nu=0}^{\infty} A_\nu G^{\nu}(z) + \sum_{\nu=1}^{\infty} zA_\nu \sum_{r=0}^{\nu-1} G^r(z)G'(z)G^{\nu-r-1}(z)\right]_{z=1}\underline{e}$$

$$= \left[G + \sum_{\nu=1}^{\infty} A_\nu \sum_{r=0}^{\nu-1} G^r G'(1) G^{\nu-r-1}\right]\underline{e}$$

$$= \underline{e} + \sum_{\nu=1}^{\infty} A_\nu \sum_{r=0}^{\nu-1} G^r \underline{\mu}.$$

Now $$\sum_{\nu=1}^{\infty} A_\nu \sum_{r=0}^{\nu-1} G^r(I-G+\tilde{G}) = \sum_{\nu=1}^{\infty} A_\nu (I-G^{\nu}+\nu\tilde{G})$$

$$= A-G+\Delta(\underline{\beta})\tilde{G}.$$

Therefore

(16) $$[I - (A-G+\Delta(\underline{\beta})\tilde{G})(I-G+\tilde{G})^{-1}]\underline{\mu} = \underline{e},$$

and after writing $I = (I-G+\tilde{G})(I-G+\tilde{G})^{-1}$ and simplifying, the desired result follows. Neuts has shown in [24], Theorem 4, that the inverse used in Formula (14) exists. Note that that μ_j equals the expected number of transitions during a first passage from the state $(i+1,j)$ to the level $\underline{i}$.

Corollary 1: The inner product $\underline{g}\ \underline{\mu}$ is given by

$$(17) \qquad \underline{g}\ \underline{\mu} = \frac{1}{1-\rho} .$$

Proof: Since $(1-\rho)\underline{g}\ \underline{\mu} = (1-\rho)\underline{g}[I-A+\tilde{G}-\Delta(\underline{\beta})\tilde{G}]^{-1}\underline{e}$ and $\underline{\pi}[I-A+\tilde{G}-\Delta(\underline{\beta})\tilde{G}] = (1-\rho)\underline{g}$, it follows that $(1-\rho)\underline{g}\ \underline{\mu} = \underline{\pi}\ \underline{e} = 1$.

Corollary 1 provides a powerful accuracy check on our numerical computations as well as having its own probabilistic significance, i.e., $\underline{g}\ \underline{\mu}$ equals the "average" number of transitions required to go from level $\underline{i+1}$ to level $\underline{i}$.

3. THE FIRST PASSAGE DISTRIBUTIONS FOR LEVELS $\underline{0}$ AND $\underline{1}$

Let us define $L_{jj'}(k)$ to be the conditional probability that starting in state $(0,j)$, the process returns to level $\underline{0}$ for the first time in state $(0,j')$ after exactly k transitions. Let $\{L(k), k\geq 1\}$ be the sequence of matrices $L(k)=\{L_{jj'}(k)\}$, which completely defines the distribution of the first passage times from level $\underline{0}$ back to level $\underline{0}$. We analogously define the sequences of matrices $\{H(k), k\geq 1\}$ and $\{K(k), k\geq 1\}$ as the first passage distributions from level $\underline{1}$ to level $\underline{0}$ and from level $\underline{1}$ to level $\underline{1}$, respectively. In the context of queueing theory $\{H(k), k\geq 1\}$ corresponds to the densities of the number of customers served during busy periods starting with one customer. We define the corresponding matrix generating functions as follows:

$$(18) \qquad L(z) = \sum_{k=1}^{\infty} L(k)z^k, \quad H(z) = \sum_{k=1}^{\infty} H(k)z^k,$$

$$K(z) = \sum_{k=1}^{\infty} K(k)z^k, \quad \text{for } 0\leq z\leq 1.$$

It follows by standard first entrance arguments, that the following equations hold:

$$(19)\qquad L(z) = zB_0 + \sum_{\nu=1}^{\infty} zB_\nu G^{\nu-1}(z)H(z),$$

$$H(z) = zC_0 + \sum_{\nu=1}^{\infty} zA_\nu G^{\nu-1}(z)H(z)$$

$$= z[I - \sum_{\nu=1}^{\infty} zA_\nu G^{\nu-1}(z)]^{-1}C_0,$$

$$K(z) = zC_0 \sum_{r=0}^{\infty} z^r B_0^r \sum_{\nu=1}^{\infty} zB_\nu G^{\nu-1}(z) + \sum_{\nu=1}^{\infty} zA_\nu G^{\nu-1}(z)$$

$$= zC_0(I-zB_0)^{-1} \sum_{\nu=1}^{\infty} zB_\nu G^{\nu-1}(z) + \sum_{\nu=1}^{\infty} zA_\nu G^{\nu-1}(z).$$

To show that the inverse in (19) exists, we see that $\sum_{\nu=1}^{\infty} zA_\nu G^{\nu-1}(z)\,\underline{e} \le \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1}\underline{e} \le (A-A_0)\underline{e}$. But under the assumed irreducibility conditions, which normally hold in practice, the matrix $(A-A_0)$ is strictly substochastic. By Corollary 2.2 in the appendix in Karlin and Taylor [10], we have that the matrix $\sum_{\nu=1}^{\infty} zA_\nu G^{\nu-1}(z)$, $0 \le z \le 1$, has spectral radius less than one and therefore the desired inverse exists. Note that the matrices $L=L(1-)$, $H=H(1-)$, $K=K(1-)$ are all stochastic, whenever G is stochastic. For example this is verified for H as follows. Since $C_0\underline{e} = A_0\underline{e}$, the vector $H\underline{e}$ may be written as

$$H\underline{e} = (I - \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1})^{-1} A_0\underline{e},$$

but clearly Equation (11) implies that $G=(I-\sum_{\nu=1}^{\infty} A_\nu G^{\nu-1})^{-1}A_0$ and therefore $H\underline{e} = G\underline{e} = \underline{e}$.

We define the invariant probability vectors $\underline{d}$ and $\underline{\kappa}$ of the matrices L and K as follows:

$$(20) \qquad \underline{d}L(1) = \underline{d} \text{ and } \underline{d}\,\underline{e} = 1,$$

$$\underline{\kappa}K(1) = \underline{\kappa} \text{ and } \underline{\kappa}\,\underline{e} = 1.$$

In the sequel, we shall derive explicit expressions for the mean vectors $\underline{d}^*$, $\underline{h}^*$ and $\underline{\kappa}^*$ defined by

$$(21) \qquad \underline{d}^* = L'(1)\underline{e}, \quad \underline{h}^* = H'(1)\underline{e}, \quad \underline{\kappa}^* = K'(1)\underline{e}.$$

Theorem 3: Provided the vector $\sum_{\nu=1}^{\infty} \nu B_\nu \underline{e}$ is finite, the mean vectors $\underline{h}^*$, $\underline{d}^*$ and $\underline{\kappa}^*$ are given by

$$(22) \qquad \underline{h}^* = (I-\sum_{\nu=1}^{\infty} A_\nu G^{\nu-1})^{-1}\left\{\underline{e}+\left[\sum_{\nu=1}^{\infty} A_\nu - \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1} + \sum_{\nu=2}^{\infty} (\nu-1)A_\nu \tilde{G}\right] (I-G+\tilde{G})^{-1}\underline{\mu}\right\},$$

$$\underline{d}^* = \underline{e}+\sum_{\nu=1}^{\infty} B_\nu G^{\nu-1}\underline{h}^* + \left[\sum_{\nu=1}^{\infty} B_\nu - \sum_{\nu=1}^{\infty} B_\nu G^{\nu-1} + \sum_{\nu=2}^{\infty} (\nu-1) B_\nu \tilde{G}\right] (I-G+\tilde{G})^{-1}\underline{\mu},$$

and

$$\underline{\kappa}^* = \underline{e} + C_0(I-B_0)^{-1}\underline{e} + \left\{C_0(I-B_0)^{-1}\left[\sum_{\nu=1}^{\infty} B_\nu - \sum_{\nu=1}^{\infty} B_\nu G^{\nu-1} + \sum_{\nu=2}^{\infty}(\nu-1)B_\nu\tilde{G}\right] + \sum_{\nu=1}^{\infty} A_\nu - \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1} + \sum_{\nu=2}^{\infty}(\nu-1)A_\nu\tilde{G}\right\}(I-G+\tilde{G})^{-1}\underline{\mu}.$$

Proof: If we differentiate $H(z)$ with respect to z we obtain

$$H'(z) = C_0 + \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1}(z)H(z) + \sum_{\nu=1}^{\infty} zA_\nu G^{\nu-1}(z)H'(z) + \sum_{\nu=2}^{\infty} zA_\nu \sum_{r=0}^{\nu-2} G^r(z)G'(z)G^{\nu-r-2}(z)H(z).$$

Letting z tend to 1- yields

$$H'(1) = H + \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1} H'(1) + \sum_{\nu=2}^{\infty} A_\nu \sum_{r=0}^{\nu-2} G^r G'(1) G^{\nu-r-2} H.$$

Therefore we have

$$\underline{h}^* = H'(1)\underline{e} = \underline{e} + \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1}\underline{h}^* + \sum_{\nu=2}^{\infty} A_\nu \sum_{r=0}^{\nu-2} G^r \underline{\mu}. \tag{23}$$

Now

$$\sum_{\nu=2}^{\infty} A_\nu \sum_{r=0}^{\nu-2} G^r(I-G+\tilde{G}) = \sum_{\nu=2}^{\infty} A_\nu(I-G^{\nu-1}+(\nu-1)\tilde{G})$$

$$= \sum_{\nu=1}^{\infty} A_\nu - \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1} + \sum_{\nu=2}^{\infty}(\nu-1)A_\nu\tilde{G}.$$

Substituting into (23) and solving for $\underline{h}^*$ yields the desired result.

Similarly,

$$\left[\frac{d}{dz} L(z)\right]_{z=1} = B_0 + \sum_{\nu=1}^{\infty} B_\nu G^{\nu-1} H + \sum_{\nu=1}^{\infty} B_\nu G^{\nu-1} H'(1)$$

$$+ \sum_{\nu=2}^{\infty} B_\nu \sum_{r=0}^{\nu-2} G^r G'(1) G^{\nu-r-2} H,$$

which implies

$$\underline{d}^* = L'(1)\underline{e} = \underline{e} + \sum_{\nu=1}^{\infty} B_\nu G^{\nu-1} \underline{h}^* + \sum_{\nu=2}^{\infty} B_\nu \sum_{r=0}^{\nu-2} G^r \underline{\mu}$$

and finally

$$\underline{d}^* = \underline{e} + \sum_{\nu=1}^{\infty} B_\nu G^{\nu-1} \underline{h}^* + \left[\sum_{\nu=1}^{\infty} B_\nu - \sum_{\nu=1}^{\infty} B_\nu G^{\nu-1} + \sum_{\nu=2}^{\infty} (\nu-1) B_\nu \tilde{G}\right] \cdot$$

$$(I-G+\tilde{G})^{-1} \underline{\mu}.$$

The formula for $\underline{\kappa}^*$ is proved analogously and the details will not be shown here.

We see that the formula for $\underline{\kappa}^*$ involves the inverse of the matrix $(I-B_0)$. If the matrix $(I-B_0)$ were singular there would exist a relabeling of the rows and columns of B_0, such that it may be written in the form

$$B_0 = \begin{bmatrix} B_0' & 0 \\ B_0'' & B_0''' \end{bmatrix} .$$

Clearly, a subset of the states $(0,1),\ldots,(0,n)$ would form

an irreducible class and the infinite matrix P would then be reducible. In the irreducible case under consideration, the matrix $(I-B_0)$ is necessarily nonsingular.

4. THE STATIONARY PROBABILITY VECTOR OF THE MATRIX P

Let x(i,j) be the limiting probability that immediately following a transition, the Markov chain is in the state (i,j). In the positive recurrent case under discussion the quantities x(i,j) form an infinite probability vector $\underline{x}$, which we will write in the partitioned form ($\underline{x}_0$, $\underline{x}_1, \underline{x}_2, \ldots$). The system of equations (3) has a unique solution with all $x(i,j) > 0$ if and only if $\rho < 1$.

Using classical arguments in the theory of Markov renewal processes, we shall derive explicit expressions for the vectors $\underline{x}_0$ and $\underline{x}_1$.

<u>Theorem 4</u>: The vectors $\underline{x}_0$ and $\underline{x}_1$ are given by

$$(24) \qquad \underline{x}_0 = \frac{\underline{d}}{\underline{d}\underline{d}^*}, \qquad \underline{x}_1 = \frac{\underline{k}}{\underline{k}\underline{k}^*}.$$

<u>Proof</u>: If we consider each transition in the infinite Markov chain P as a discrete time step, then the times between successive visits to the states (0,1),...,(0,n) and the states visited, define a Markov renewal process with n states. The sojourn times in this n-state Markov renewal process are lattice random variables and the transition probability matrix of the process is given in an

equivalent form by the matrix of generating functions $L(z)$. There is a classical theorem, see e.g. Hunter [9], p. 196, in the theory of Markov renewal processes that states that the mean recurrence time E_j of a particular state $(0,j)$ is given by

$$(25)\qquad E_j = \frac{1}{d_j} \sum_{\nu=1}^{n} d_\nu d_\nu^* , \quad \text{for } 1 \le j \le n,$$

where d_ν and d_ν^* are the ν-th components of the vectors $\underline{d}$ and $\underline{d}^*$ respectively.

The mean recurrence time E_j in this finite state Markov renewal process is none other than the expected number of transitions between successive returns to the state $(0,j)$ in the infinite state Markov chain P. Clearly, we see that the stationary probability $x(0,j)$ is given by

$$(26)\qquad x(0,j) = \frac{1}{E_j} = \frac{d_j}{\sum_{\nu=1}^{n} d_\nu d_\nu^*} , \quad \text{for } 1 \le j \le n,$$

or in vector notation,

$$\underline{x}_0 = \frac{\underline{d}}{\underline{d}\underline{d}^*} .$$

To derive an explicit expression for $\underline{x}_1$, we consider the Markov renewal process defined by the times between successive visits to the states $(1,1),\ldots,(1,m)$ and the states visited. The transition probability matrix for this process

is given in an equivalent form by the matrix of generating functions, K(z). Using a completely analogous argument, we see that Equation (24) for $\underline{x}_1$ holds.

Since the matrices L and K are known in terms of matrices which may be computed explicitly, Formula (24) yields the vectors $\underline{x}_0$ and $\underline{x}_1$ in a tractable form. Knowing that the vector $\underline{x}$ must satisfy Equation (3), we must show that the following relationship between $\underline{x}_0$ and $\underline{x}_1$ hold.

Corollary 2: The vector $\underline{x}_0$ is related to the vector $\underline{x}_1$ by the following equality:

$$
\begin{aligned}
(27) \qquad \underline{x}_0 &= \underline{x}_0 B_0 + \underline{x}_1 C_0 \\
&= \underline{x}_1 C_0 (I-B_0)^{-1}.
\end{aligned}
$$

Proof: By a lengthy, but straightforward calculation given in Appendix I.

We see that Equation (27) provides us with yet another accuracy check on our numerical computations.

5. THE DERIVATIVES OF THE PERRON-FROBENIUS EIGENVALUE

In deriving the moments of the stationary distribution, we will need explicit expressions for the derivatives of the Perron-Frobenius eigenvalues and the associated eigenvectors of the matrix $A^*(z)$, defined in (6). In this section, we derive the necessary recurrence relations needed in the computation of these derivatives.

For $z \leq 1$, the matrix $A^*(z)$ has a uniquely defined Perron-Frobenius eigenvalue $\delta(z)$. Let $\underline{u}(z)$ and $\underline{v}(z)$ be the corresponding right and left eigenvectors, respectively, such that the normalizing conditions

$$(28) \qquad \underline{v}(z)\underline{u}(z) = \underline{v}(z)\underline{e} = 1,$$
$$\underline{v}(1) = \underline{\pi}, \text{ and } \underline{u}(1) = \underline{e},$$

hold in addition to the defining relations

$$(29) \qquad [A^*(z)-\delta(z)I]\underline{u}(z) = \underline{v}(z)[A^*(z)-\delta(z)I] = \underline{0}.$$

We denote by $A^*_\nu(z)$, the matrix obtained by differentiating each entry of $A^*(z)$, ν times.

Theorem 5: The derivatives $\delta^{(n)}(1)$, $\underline{u}^{(n)}(1)$, $\underline{v}^{(n)}(1)$, $n \geq 0$, may be computed recursively for each n for which $A^*_n(1)$ is finite. The recursion formulas are

$$\delta^{(0)}(1) = 1,\ \underline{u}^{(0)}(1) = \underline{e},\ v^{(0)}(1) = \underline{\pi} \tag{30}$$

$$\delta^{(1)} = \underline{\pi}A_1^*(1)\underline{e} = \rho$$

$$\underline{u}^{(1)}(1) = (I-A+\Pi)^{-1}[A_1^*(1)-\delta^{(1)}(1)I]\underline{e}$$
$$= (I-A+\Pi)^{-1}\underline{\beta} - \rho\underline{e},$$

$$\underline{v}^{(1)}(1) = \underline{\pi}[A_1^*(1)-\delta^{(1)}(1)I](I-A+\Pi)^{-1}$$
$$= \underline{\pi}A_1^*(1)(I-A+\Pi)^{-1} - \rho\underline{\pi},$$

and for $n\geq 2$

$$\delta^{(n)}(1) = \sum_{\nu=1}^{n} \binom{n}{\nu}\underline{\pi}A_\nu^*(1)\underline{u}^{(n-\nu)}(1) - \sum_{\nu=1}^{n-1} \underline{\pi}\underline{u}^{(n-\nu)}(1)\,\delta^{(\nu)}(1) \tag{31}$$

$$\underline{u}^{(n)}(1) = (I-A+\Pi)^{-1}\sum_{\nu=1}^{n}\binom{n}{\nu}\left[A_\nu^*(1)-\delta^{(\nu)}(1)I\right]\underline{u}^{(n-\nu)}(1) - \left[\sum_{\nu=1}^{n-1}\binom{n}{\nu}\underline{v}^{(\nu)}(1)\underline{u}^{(n-\nu)}(1)\right]\underline{e}$$

$$\underline{v}^{(n)}(1) = \sum_{\nu=0}^{n-1}\binom{n}{\nu}\underline{v}^{(\nu)}(1)\left[A_{n-\nu}^*(1)-\delta^{(n-\nu)}(1)I\right](I-A+\Pi)^{-1}$$

where Π is the square matrix of order m, whose rows are all identical and are equal to the vector $\underline{\pi}$.

<u>Proof</u>: The values corresponding to n=0 are obvious. By differentiating n times in Formula (29), we obtain

(32) $$\sum_{\nu=0}^{n} \binom{n}{\nu} [A_\nu^*(z)-\delta^{(\nu)}(z)I]\,\underline{u}^{(n-\nu)}(z) = \underline{0}.$$

Premultiplying (32) by $\underline{v}(z)$, letting z tend to 1 and rearranging terms leads to

(33) $$\delta^{(n)}(1) = \sum_{\nu=1}^{n} \binom{n}{\nu} \underline{\pi} A_\nu^*(1)\,\underline{u}^{(n-\nu)}(1) - \sum_{\nu=1}^{n-1} \underline{\pi}\,\underline{u}^{(n-\nu)}(1)\,\delta^{(\nu)}(1).$$

Letting z tend to 1 in Equation (32) and rearranging terms leads to

(34) $$(I-A)\underline{u}^{(n)}(1) = \sum_{\nu=1}^{n} \binom{n}{\nu} \left[A_\nu^*(1)-\delta^{(\nu)}(1)I\right]\underline{u}^{(n-\nu)}(1),$$

which is a singular system of equations. Adding $\Pi\underline{u}^{(n)}(1)=[\underline{\pi}\,\underline{u}^{(n)}(1)]\underline{e}$ to both sides and noting that $(I-A+\Pi)^{-1}\underline{e} = \underline{e}$, we obtain

(35) $$\underline{u}^{(n)}(1) = (I-A+\Pi)^{-1} \sum_{\nu=1}^{n} \binom{n}{\nu} \left[A_\nu^*(1)-\delta^{(\nu)}(1)I\right]\underline{u}^{(n-\nu)}(1) + (\underline{\pi}\,\underline{u}^{(n)}(1))\underline{e}.$$

In order to determine $\underline{u}^{(n)}(1)$ in terms of earlier terms of the recurrence, we differentiate n times in the normalizing condition $\underline{v}(z)\underline{u}(z)=1$, and let z tend to 1 to obtain

(36) $$\underline{\pi}\,\underline{u}^{(n)}(1) = -\sum_{\nu=1}^{n} \binom{n}{\nu} \underline{v}^{(n)}(1)\underline{u}^{(n-\nu)}(1).$$

Note that for n=1, $\underline{\pi u}^{(1)}(1) = 0$, and

(37) $$\underline{v}^{(n)}(1)\underline{u}(1) = \underline{v}^{(n)}(1)\underline{e} = 0.$$

Substitution of (36) into (35) yields the stated formula for $\underline{u}^{(n)}(1)$. The vectors $\underline{v}^{(n)}(1)$ are obtained by differentiating the second equation in (29) n times and setting z=1. We get

(38) $$\underline{v}^{(n)}(1)(I-A) = \sum_{\nu=0}^{n=1} \binom{n}{\nu}\underline{v}^{(\nu)}(1)\left(A^*_{n-\nu}(1) - \delta^{(n-\nu)}(1)\, I\right),$$

but since $\underline{v}^{(n)}(1)\Pi = [\underline{v}^{(n)}(1)\underline{e}]\pi = \underline{0}$, we have

$$\underline{v}^{(n)}(1) = \sum_{\nu=0}^{n-1} \binom{n}{\nu}\underline{v}^{(\nu)}(1)\left(A^*_{n-\nu}(1) - \delta^{(n-\nu)}(1)\, I\right)(I-A+\Pi)^{-1}.$$

Setting n=1, we obtain the stated explicit formulas.

6. THE MOMENTS OF THE STATIONARY DISTRIBUTION

In the next section, we shall develop a recursive algorithm to compute further components of the vector $\underline{x}$. As a criterion for truncation in the infinite system of equations $\underline{x}P = \underline{x}$, we will use the moments of the stationary distribution. We presently derive complicated yet tractable expressions for these moments.

If we let z tend to 1 in Equation (7), we get

$$(39) \qquad \underline{X}(1)(I-A) = \underline{x}_0 \sum_{k=1}^{\infty} B_k - \underline{x}_1 A_0.$$

Adding $\underline{X}(1)\Pi = (\underline{X}(1)\underline{e})\underline{\pi} = (1-\underline{x}_0\underline{e})\underline{\pi}$ to both sides of Equation (39) and recognizing that $\underline{\pi}(I-A+\Pi)^{-1} = \underline{\pi}$, we have

$$(40) \qquad \underline{X}(1) = \left[\underline{x}_0 \sum_{k=1}^{\infty} B_k - \underline{x}_1 A_0\right](I-A+\Pi)^{-1} + (1-\underline{x}_0\underline{e})\underline{\pi}.$$

We see that we may calculate the vector $\underline{X}(1)$ in terms of the data and the known vectors $\underline{x}_0$ and $\underline{x}_1$. This gives us an accuracy check on the numerical computations of the additional components of the vector $\underline{x}$. Having computed the vectors $\underline{x}_2, \ldots, \underline{x}_k$, the sum $\sum_{\nu=1}^{k} \underline{x}_\nu$ should be entrywise close to $\underline{X}(1)$. By evaluating the n-th derivative $\underline{X}^{(n)}(1)$, we

obtain $\pi_j^{-1} X_j^{(n)}(1)$ the n-th conditional factorial moment of the stationary distribution given that immediately following a transition the process is in state (i,j), for some i≥1. The quantity $X'(1)\underline{e}$ is the n-th factorial moment of the stationary distribution.

Theorem 6: The vectors $\underline{X}^{(1)}(1)$ and $\underline{X}^{(2)}(1)$ are given by

$$(41) \quad \underline{X}^{(1)}(1) = \left\{-\underline{X}(1)\left[I-\sum_{k=1}^{\infty} kA_k\right] + \underline{x}_0 \sum_{k=1}^{\infty} B_k + \underline{x}_0 \sum_{k=1}^{\infty} kB_k - \underline{x}_1 A_0\right\}(I-A+\Pi)^{-1} + (\underline{X}^{(1)}(1)\underline{e})\underline{\pi}$$

where

$$(42) \quad \underline{X}^{(1)}(1)\underline{e} = \frac{1}{2(1-\rho)}\left\{2\underline{x}_0 \sum_{k=1}^{\infty} kB_k\underline{e} + 2\underline{x}_0 \sum_{k=1}^{\infty} B_k\underline{u}^{(1)}(1) + \underline{x}_0 \sum_{k=2}^{\infty} k(k-1)B_k\underline{e} + 2\underline{x}_0 \sum_{k=1}^{\infty} kB_k\underline{u}^{(1)}(1) + \underline{x}_0 \sum_{k=1}^{\infty} B_k\underline{u}^{(2)}(1) - 2\underline{x}_1 A_0\underline{u}^{(1)}(1) - \underline{x}_1 A_0\underline{u}^{(2)}(1) + \underline{X}(1)\underline{e}\delta^{(2)}(1)\right\} - \underline{X}(1)\underline{u}^{(1)}(1),$$

and

$$\underline{X}^{(2)}(1) = \left\{ \underline{X}(1) \sum_{k=2}^{\infty} k(k-1)A_k - 2\underline{X}^{(1)}(1)\left[I - \sum_{k=1}^{\infty} kA_k\right] + 2\underline{x}_0 \sum_{k=1}^{\infty} kB_k + \underline{x}_0 \sum_{k=2}^{\infty} k(k-1)B_k \right\}(I-A+\Pi)^{-1} + (\underline{X}^{(2)}(1)\underline{e})\underline{\pi}, \tag{43}$$

where

$$\begin{aligned}\underline{X}^{(2)}(1)\underline{e} = \frac{1}{3(1-\rho)} \Bigg\{ & 3\underline{X}^{(1)}(1)\underline{e}\delta^{(2)}(1) \\ & + 3\underline{X}(1)\underline{u}^{(1)}(1)\delta^{(2)}(1) + \underline{X}(1)\underline{e}\delta^{(3)}(1) \\ & + 3\underline{x}_0 \sum_{k=2}^{\infty} k(k-1)B_k\underline{e} + 6\underline{x}_0 \sum_{k=1}^{\infty} kB_k\underline{u}^{(1)}(1) \\ & + 3\underline{x}_0 \sum_{k=1}^{\infty} B_k\underline{u}^{(2)}(1) + \underline{x}_0 \sum_{k=3}^{\infty} k(k-1)(k-2)B_k\underline{e} \\ & + 3\underline{x}_0 \sum_{k=2}^{\infty} k(k-1)B_k\underline{u}^{(1)}(1) \\ & + 3\underline{x}_0 \sum_{k=1}^{\infty} kB_k\underline{u}^{(2)}(1) + \underline{x}_0 \sum_{k=1}^{\infty} B_k\underline{u}^{(3)}(1) \\ & - 3\underline{x}_1 A_0\underline{u}^{(2)}(1) - \underline{x}_1 A_0\underline{u}^{(3)}(1) \Bigg\} \\ & - \frac{4}{3}\underline{X}^{(1)}(1)\underline{u}^{(1)}(1) - \underline{X}(1)\underline{u}^{(2)}(1).\end{aligned} \tag{44}$$

Proof: The lengthy derivations are shown in Appendix II.

Note that although the formulas for the conditional factorial moments are complicated, they involve only known quantities and are in a computationally tractable form.

7. AN ITERATIVE METHOD FOR THE COMPUTATION OF THE COMPONENTS OF THE STATIONARY VECTOR $\underline{x}$

We recall that the components of the vector $\underline{x}$ satisfy the equations

$$(45) \qquad \underline{x}_k = \underline{x}_0 B_k + \sum_{\nu=1}^{k+1} \underline{x}_\nu A_{k+1-\nu}, \quad \text{for } k \geq 1.$$

We see that if the matrix A_0 is nonsingular, the vector $\underline{x}_{k+1}$, $k \geq 1$, may be found by solving the appropriate equation in (45). Neuts has shown in [27], that in the case where A_0 is singular, the vectors $\underline{x}_{k+1}$, $k \geq 1$, may, in principle, still be computed recursively if the rank of the matrix $A_0(I-A_1)^{-1}$ is equal to the rank of the matrix $[A_0(I-A_1)^{-1}]^2$. This recursive procedure, however, is, except in very special cases, numerically highly unstable.

We suggest computing the vectors $\underline{x}_k$, $k \geq 1$, using the following block Gauss-Seidel iterative procedure.

$$(46) \qquad \underline{x}_k(0) = \underline{b}_k'$$

$$\underline{x}_k(n+1) = \underline{b}_k' + \sum_{\nu=2}^{k-1} \underline{x}_\nu(n+1) A_{k+1-\nu}' + \underline{x}_{k+1}(n) A_0'$$

where $\underline{b}_k' = (\underline{x}_0 B_k + \underline{x}_1 A_k)(I-A_1)^{-1}$ and $A_\nu' = A_\nu (I-A_1)^{-1}$.

Note that the vectors $\underline{b}_k'$ are now known quantities. Note also that in the recurrence relationship for $\underline{x}_k$, we use the most recent iterates of $\underline{x}_\nu$, $\nu=2,\ldots,k-1$. Using the moments that we have computed in the last section, we first truncate the infinite system of equations, (46), at some index k*, where k* is the smallest integer not less than $\mu + 3\sigma$. (μ and σ being the mean and standard deviation of the queue length following departures, respectively.) We continue the iterations until the condition

$$\max_{2\le k\le k^*} \left\{ \underline{x}_k(n) - \underline{x}_k(n-1) \right\} < 10^{-8} \tag{47}$$

is reached. At this time we check to see if an adequate number of components have been computed. This amounts to computing ε, where

$$\varepsilon = 1 - \sum_{k=0}^{k^*} \underline{x}_k(n)\underline{e} \quad . \tag{48}$$

If $\varepsilon > 10^{-4}$ we increase k* by 1 and continue with the iterations. When all of the conditions for stopping have been reached, we utilize an accuracy check on the components of the vectors $\underline{x}_k$, $2\le k\le k'$, where k' is the number of components computed. From Equation (5), we have

$$\underline{X}(1) = \sum_{\nu=1}^{\infty} \underline{x}_\nu, \tag{49}$$

where $\underline{X}(1)$ is known explicitly.

8. APPLICATIONS

A. The $M^X/G/1$ Queue

We consider a single server queue with a general service time distribution and arrivals of random group sizes, which occur at the epochs of a Poisson process. It is well-known that the successive queue lengths immediately following departures in such a queue (denoted by $M^X/G/1$) form a Markov chain of type (1) where the matrices are all scalars. In this case, the first two rows are identical and the entry a_ν corresponds to the probability that there are ν arrivals during the service of one customer. For the scalar case,

$$I = G = \tilde{G} = A = 1, \text{ and } \beta = \rho = \frac{\lambda\xi}{\alpha} ,$$

where λ is the mean arrival rate, ξ is the mean group size and α is the mean service rate. Formula (14) for μ then simplifies to

$$\mu = \frac{1}{1-\rho} ,$$

which is the classical formula for the mean number of services during a busy period.

B. Two Queues in Series with Finite Intermediate Waitingroom

The following queueing model has been studied by several authors [18,19,35]. A system of queues consists of two units. Customers arrive at a first unit (I) according to a homogeneous Poisson process of rate λ. The service times in unit I are independent, identically distributed random variables with common distribution function $H(\cdot)$. We also assume that $H(\cdot)$ has a positive finite mean.

Upon completion of service in unit I, all customers go on to a second unit (II) via a finite waitingroom. We assume that there cannot be more than k customers in unit II and in the waitingroom at any time. If upon completion of service in unit I a customer finds the waitingroom full, then the unit one "blocks until a service in unit II is completed. At that time he is allowed to enter the waitingroom.

We assume that the service times in unit II are independent, identically distributed random variables with a negative exponential distribution. The service times in unit II are also stochastically independent of those in unit I and of the arrival process. If we look at the number of customers in the system immediately following a service in unit I, we have an embedded Markov chain P of the type (1) and state space $\{(i,j)\ i \geq 0,\ 1 \leq j \leq k+1\}$ where i is the number

of customers in the system, who have not yet completed service in unit I and j is the number of customers in the system, who have completed service in unit I, but not yet in unit II. The states for which j=k+1 correspond to blocking.

In this model, m=n. The specific form of the matrices A_ν and B_ν, $\nu \geq 0$, is complicated, but is readily deduced from Formulas (3) - (18) in [18]. It should be noted that most of the analysis depends only on the prevailing special structure of the matrix P and not on the complexities of the precise definitions of the matrices A_ν and B_ν.

C. A Single Server Queue with Versatile Markovian Input

In [30], Neuts defined a general class of Markovian point processes, which generalize the classical Poisson process and also renewal processes of phase type [23]. Such point processes are useful in modelling a large number of qualitative features of arrival processes. Among these are group arrivals, randomly fluctuating arrival rates and inhibitory phenomena.

In his thesis [34], V. Ramaswami has extended the theory of the simple M/G/1 queue to a single server queueing model having this versatile Markovian point process as

its input and general independent, identically distributed service times.

Although the detailed definition of the matrices A_ν and B_ν is again highly involved and will not be repeated here, the queueing model studied by Ramaswami has an embedded Markov chain of the type (1). Once the vector $\underline{x}$ has been evaluated, which may be done by using the techniques proposed here, one can then draw upon the detailed results in [34] to compute a large number of other qualitative queue features, such as the steady-state distributions of the virtual waiting time and the queue length at an arbitrary point in time.

D. Queues with Exceptional Services

Consider a queueing situation in which there are occasionally "exceptional" services. For example, the service mechanism may occasionally break down, after which there may be a certain amount of time needed for repair before a service can be performed. We can consider the breakdown and repair time combined as forming an exceptional service. In order to formulate this model, we shall need some of the basic properties of phase type distributions (PH-distributions) and renewal processes of phase type (PH-Renewal Processes), which were introduced by M.F. Neuts [22, 23]. Only the basic definitions of PH-distributions will

be reviewed here. The interested reader is referred to the cited references for further details.

Consider an (m+1)-state Markov chain on the integers $\{1,\ldots,m,m+1\}$, whose matrix P of stationary transition probabilities is of the form

$$P = \begin{bmatrix} T & \underline{T}^{\circ} \\ \underline{0} & 1 \end{bmatrix},$$

where T is an $m\times m$ matrix and $\underline{T}^{\circ}$ is a column vector with m components. We shall assume that the probability of absorption into the state m+1, starting from any given initial state, is equal to one. This implies that $(I-T)^{-1}$ exists. The vector of initial probabilities of the Markov chain will be denoted by $(\underline{\alpha},\alpha_{m+1})$ and here we may assume that $\alpha_{m+1}=0$.

A probability density $\{r_k\}$ on the positive integers is of <u>phase type</u>, if and only if there exists a finite stochastic matrix P of the type (1) and a vector $\underline{\alpha}$ of initial probabilities, such that $\{r_k\}$ is the density of the time till absorption into the state m+1. If $\{r_k\}$ is of phase type, then it is easily seen that

$$r_k = \underline{\alpha}T^{k-1}\underline{T}^{\circ}, \quad \text{for } k\geq 1.$$

Since the density $\{r_k\}$ is determined by $\underline{\alpha}$ and T, we call the

pair $(\underline{\alpha},T)$ a representation of the density $\{r_k\}$.

Now consider the stochastic matrix Q, of order m, defined by

$$Q = T + T^{\circ}A^{\circ},$$

where $T^{\circ}_{ij} = T^{\circ}_{i}$, for $1 \leq i$, $j \leq m$, and $A^{\circ} = \mathrm{diag}(\alpha_1,\ldots,\alpha_m)$. The matrix Q is readily shown to be the transition matrix of the PH-renewal process obtained by instantaneously restarting the chain P after each absorption by performing a multinomial trial with probabilities $\alpha_1,\ldots,\alpha_m$, to select the new "initial" state. Considering each absorption as a renewal, it is obvious that the density of the times between renewals is of phase type and is equal to $\{r_k\}$.

We can now construct a model of a queue with exceptional services. Consider a queue with exponential interarrival times with arrival rate λ. We introduce an underlying m-state discrete PH-renewal process as defined above such that immediately prior to a service completion, a transition is made in the PH-renewal process. If this transition does not involve a renewal then the service time of the next service has the distribution $F(x)$. If the transition involves a renewal, then the service time of the next service has the distribution $F_1(x)$. In this way, the exceptional services correspond to renewals in the discrete

PH-renewal process. If we define J_n to be the phase of the PH-renewal process during the n^{th} service and X_n to be the duration of the n^{th} service then the pair (J_n, X_n) form a Markov renewal process with transition probabilities given by

$$Q_{ij}(x) = P\{J_{n+1}=j,\ X_n \le x \mid J_n = i\}$$
$$= T_{ij}F(x) + T_i^{\circ}\alpha_j F_1(x).$$

We define the matrix $Q(x) = \{Q_{ij}(x)\} = TF(x) + T^{\circ}A^{\circ}F_1(x)$.

The successive queue lengths immediately following departures and the random variables J_n form a Markov chain of the type (1), where $C_0 = A_0$, $B_n = A_n$ for $n=0,1,\ldots,$ and the matrices A_n are defined by

$$A_n = \int_0^{\infty} e^{-\lambda u} \frac{(\lambda u)^n}{n!}\, dQ(x).$$

The above model may easily be modified to allow for a different arrival rate during the exceptional services as may be the case in certain practical situations.

We see that in this manner a generalization of the M/G/1 arises in which strings of ordinary services are separated by single exceptional services. The lengths of the runs of ordinary services are independent, identically distributed random variables, which may have an arbitrary distribution of phase type. Mathematically the queue so

obtained is a particular case of the M/SM/1 queue. It is computationally highly tractable and permits the algorithmic investigation of a number of control and optimization aspects, which we shall discuss elsewhere.

E. Bulk Service Queueing Models

1. Bailey's Bulk Service Queue. Consider a bulk queueing model involving a server, who becomes available at the epochs of a renewal process with underlying distribution $H(\cdot)$. Customers arrive according to a Poisson process of rate λ. If k customers are present when the server becomes available, a group of size min(k,m) enters service. This model was solved by N.T.J. Bailey [1] by the use of complex variable methods.

The successive queue lengths immediately prior to the beginnings of services form a Markov chain P with the following structure:

$$P = \begin{array}{c|cccccc|cc|}
0 & a_0 & a_1 & a_2 & a_3 & \cdots & a_m & a_{m+1} & \cdots \\
1 & a_0 & a_1 & a_2 & a_3 & \cdots & a_m & a_{m+1} & \cdots \\
2 & a_0 & a_1 & a_2 & a_3 & \cdots & a_m & a_{m+1} & \cdots \\
\vdots & \vdots & & & & & & & \\
m & a_0 & a_1 & a_2 & a_3 & \cdots & a_m & a_{m+1} & \cdots \\
\cline{2-9}
m+1 & a_0 & a_1 & a_2 & a_3 & \cdots & a_m & a_{m+1} & \cdots \\
m+2 & 0 & a_0 & a_1 & a_2 & \cdots & a_{m-1} & a_m & \cdots \\
m+3 & 0 & 0 & a_0 & a_1 & \cdots & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & & & &
\end{array}$$

where $a_j = \int_0^\infty e^{-\lambda u} \frac{(\lambda u)^j}{j!}\, dH(u)$, for $j \geq 0$. We see that this matrix may be partitioned into the form (1), where $C_0 = A_0$ and all of the matrices B_n and A_n are square matrices of order m. We also note some interesting consequences of the fact that the first m rows are identical and given by $\{a_j\}$. We state these results in the following theorem which may be found in Neuts [25].

<u>Theorem 7</u>. For the Markov chain in Bailey's model, the matrix $L(z)$ defined in Formula (18), has m identical rows which are equal to te first row of $G(z)$, defined in Formula (8). The vector $\underline{d}$, defined in (20) is given by the first

row of the matrix G and the vector $\underline{d}^*$, defined in (21), is given by

$$\underline{d}^* = \mu_1 \underline{e},$$

where μ_1 is the first component of the vector $\underline{\mu}$. The vector $\underline{x}_0$ is given by

$$\underline{x}_0 = \mu_1^{-1} \underline{d},$$

and

$$\underline{x}_0 \underline{e} = \mu_1^{-1}.$$

2. Moran's Dam with Infinite Capacity. A classical model, due to Moran [13], for a dam in discrete time with discretized content, involves a Markov chain P of the type (1), with the following entries:

$$C_0 = A_0$$

$$\{A_0\}_{ij} = a_{j-i}, \text{ if } j \geq i, \quad 1 \leq i, \ j \leq m$$
$$= 0, \text{ if } j < i$$

$$\{A_\nu\}_{ij} = a_{\nu m+j-i},$$

$$\{B_0\}_{j0} = \sum_{\nu=0}^{m-j} a_\nu,$$

$$\{B_0\}_{jk} = a_{k+m-j}, \quad k \geq 1,$$

$$\{B_\nu\}_{ij} = a_{\nu m+m-i}, \quad \nu \geq 1,$$

where $\{a_k\}$ is the probability density of the number of units of water added to the dam per year and m is the maximum amount of water released at the end of each year. We assume that the capacity of the dam is infinite.

3. A Bulk Service Queue with a Threshold. The following queueing model also has an embedded Markov chain of the type described in this thesis. Customers arrive at a service unit according to a Poisson process of rate λ. Services occur in groups, with the group size dependent on the the queue length according to the following rule. Let there be i customers waiting at the completion of a service. If $0 \leq i < L$, the server remains idle until the queue length reaches L and then starts serving all L customers. If $L \leq i \leq m$, a group of size i enters service and if $i \geq m$, a group of size m is served. It is assumed that the lengths of service of successive groups are conditionally independent, given the group size. The successive queue lengths following departures form a Markov chain of the desired structure. This model has been studied by several authors [5,14,17,25].

4. A Bulk Service Queue Viewed as a Branching Process. Assume that a server serves groups of size m. If at time t=0, there are i customers we divide this group into groups of size m, with any remaining customers left alone. Assume that there are n such groups of size m. We consider

any arrivals to the queue during the service of any of these groups as the progeny of that group. The total number of customers at the end of the service of the n-th group will form the first generation. If we continue in this manner then the busy period starting with i customers will be equal to the time till extinction in this branching process starting with i customers. This model has been studied by Ezhov and Shakhbazov [6].

F. Queues with Semi-Markov Service Times

Queues with semi-Markov service times have been studied by several authors including Çinlar, Gaver, Loynes, Neuts and P. Purdue [2,7,12,15,27,29,33]. One typical model involves an M/G/1 queue in which there are m types of customers, operating under the first come-first served discipline. We assume that the server expends a random length of time in the change-over from one type of customer to another. This model has an embedded Markov chain with state space $\{(i,j),\ i\geq 0,\ 1\leq j\leq m\}$ where i is the number of customers in the queue following a service and j is the type of service that the server is tooled up for immediately following a service. etails of this model are given in [27]. Explicit formulas for general M/SM/1 queues with group arrivals are available in [29].

G. Generalized Random Walks

Consider an infinite Markov chain with state space $\{(0,j),\ 1\le j\le n, \text{ and } (i,j),\ i\ge 0,\ 1\le j\le m\}$. We assume that starting in state (i,j) $i\ge 1$, $1\le j\le m$, in one transition, the chain may only enter the states $\{(i,j'), j'\ne j, 1\le j'\le m$, and $(i-1,j'), (i+1,j'), 1\le j'\le m\}$. Starting in the state $(0,j)$, only the states $\{(0,j'), j'\ne j, 1\le j'\le n$ and $(1,j'), 1\le j'\le m\}$ may be reached. This model describes a generalized random walk and has a transition probability matrix P of the form

$$P = \begin{bmatrix} B_0 & B_1 & 0 & 0 & 0 & 0 & \cdots \\ C_0 & A_1 & A_2 & 0 & 0 & 0 & \cdots \\ 0 & A_0 & A_1 & A_2 & 0 & 0 & \cdots \\ 0 & 0 & A_0 & A_1 & A_2 & 0 & \cdots \\ 0 & 0 & 0 & A_0 & A_1 & A_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{bmatrix},$$

which is clearly of the type (1). This model has been studied in references [33,37,39].

The invariant probability vector of this process may be evaluated using the techniques proposed in this paper. We also note that the matrix P has a structure, which is a special case of a general class of Markov chains

studied by Neuts [26]. It is shown there that the invariant vector is of a matrix geometric form.

H. Queues with Fluctuating Input and Service

Consider a queue where the arrival process and/or the service rate exhibit random variations. This model may be used to describe changes in work shifts, rush hours, interruptions in the arrival process, server breakdowns, etc. To be specific, we assume that there is an underlying m-state continuous parameter irreducible Markov chain which governs the phases. During any interval spent in phase i, the arrivals are according to a homogeneous Poisson process of rate λ_i and any service initiated during such an interval has a duration distributed according to $H_i(\cdot)$. This model was first discussed by Neuts [20]. If we consider the queue lengths immediately following departure we see that once again, we have an embedded Markov chain of the type (1).

By making Markovian assumptions on the service mechanism, it is also possible to consider models for which the service rate of a customer may vary with the underlying phase state. This model, usually called "The M/M/c queue in a Markovian environment," has an extensive literature [3,28,31,32,38]. It may also be treated by the methods described in this thesis, although results which identify

the stationary probability vector as being (modified) matrix-geometric, provide a more explicit solution, both from an analytic and an algorithmic point of view.

I. Parity Dependent Service Times

A model which is related to the above "bin" model is a queue in which customers are served one at a time but the service time depends on the parity (residue class) of the queue. If we consider the residue class of m=3, we have the following embedded Markov chain:

$$P = \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ \vdots \end{array} \left[\begin{array}{ccc|ccc|cc} a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & \cdots \\ \hline a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & \cdots \\ 0 & a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & \cdots \\ 0 & 0 & a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & \cdots \\ \hline 0 & 0 & 0 & a_{10} & a_{11} & a_{12} & a_{13} & \cdots \\ 0 & 0 & 0 & 0 & a_{20} & a_{21} & a_{22} & \cdots \\ 0 & 0 & 0 & 0 & 0 & a_{30} & a_{31} & \cdots \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & a_{10} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{array} \right]$$

We see that we may partition this matrix exactly as in the "bin" model to obtain a matrix of type (1). This model was

studied by Neuts [15]. We note that the first row of the matrix P_2 in [15] should consist of the quantities $\{a_\nu\}$, rather than $\{b_\nu\}$. This also results in some obvious changes in Section 3 of that paper.

J. A Bin Model

Consider the following "bin" model. The waiting room of a queueing system consists of an unlimited number of bins of size m. As arrivals occur, they are placed into the bins, filling them one at a time. When the server becomes available, he serves the last bin that received input, if it is not full. If all occupied bins are full, he chooses one at random and serves it. Now suppose that the service times depend on the number of items in the bin being served (i.e. the residue class of the queue). The queue length immediately following a departure forms a Markov chain P with the following structure. (For simplicity, we display the matrix for m=3).

$$P = \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \left| \begin{array}{ccc|ccc|cc} a_{10} & a_{11} & a_{12} & a_{13} & . & . & . & \\ \hline a_{10} & a_{11} & a_{12} & a_{13} & . & . & . & \\ a_{20} & a_{21} & a_{22} & a_{23} & . & . & . & \\ a_{30} & a_{31} & a_{32} & a_{33} & . & . & . & \\ \hline 0 & 0 & 0 & a_{10} & a_{11} & a_{12} & a_{13} & \cdots \\ 0 & 0 & 0 & a_{20} & a_{21} & a_{22} & a_{23} & \cdots \\ 0 & 0 & 0 & a_{30} & a_{31} & a_{32} & a_{33} & \cdots \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & a_{10} & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{20} & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{30} & \cdots \end{array} \right|$$

The above matrix is also of the type (1).

9. THE APL PROGRAM

The algorithms constructed in this paper have been programmed and implemented in APL. We briefly review the purpose of each of the APL functions and give a short discussion of the more important ones. In Section 10, we present numerical examples.

The index origin of the workspace is set to zero. The sequences of matrices $\{A_n\}$ and $\{B_n\}$ have been truncated at some index k. We denote the matrix $A=\sum_{\nu=0}^{k} A$ at the variable AA. The sequence of matrices $\{A_n\}_{n=0}^{k}$ is represented by the three dimensional array A of dimensions (k+1,n,m). The variables CNOT and BNOT equal the matrices C_0 and B_0 respectively. The sequence of matrices $\{B_n\}_{n=1}^{k}$ is the three dimensional array B of dimension (k,n,m). Note that in the following program, $A[k;;] = A_k$ and $B[k;;] = B_{k+1}$, for $k \geq 0$.

A. The APL Functions

```
      ∇   RUN
[1]       TIME←⍳21
[2]       ⎕IO←0
[3]       M←1↑⍴CNOT
[4]       N←1↓⍴CNOT
[5]       MM←1↑⍴A
[6]       FI←STVECT AA
[7]       BETA←+/A1←A1MEAN
[8]       RHO←FI+.×BETA
[9]       →(RHO<0.99)/CONTINUE
[10]      '*** RHO= ';RHO;' **** '
[11]      '      TRY AGAIN!'
```

```
  →0
CONTINUE:G←SOLVG
 MU←MUVECT
 →((GVECT+.xMU)=1÷1-RHO)/CONT1
 'ACCURACY CHECK 1 FAILED!!'
 ''
 'THE INNER PRODUCT OF THE VECTORS G AND MU IS:'
 ''
 'BF12.10'$GVECT+.xMU
 ''
 'THE QUANTITY ONE DIVIDED BY ONE MINUS RHO IS:'
 ''
 'BF12.10'$1÷1-RHO
 ''
CONT1:L←LMATRIX
 K←KMATRIX
 D←STVECT L
 KAPPA←STVECT K
 DSTAR←SOLVL
 KSTAR←SOLVK
 XNOT←D÷D+.xDSTAR
 XONE←KAPPA÷KAPPA+.xKSTAR
 →(XNOT=(XNOT+.xBNOT)+XONE+.xCNOT)/CONT2
 'ACCURACY CHECK 2 FAILED!!'
 ''
 'XNOT EQUALS:'
 ''
 XNOT
 ''
 'THE QUANTITY:'
 ''
 '    XNOT TIMES BNOT PLUS XONE TIMES CNOT'
 ''
 'EQUALS:'
 ''
 (XNOT+.xBNOT)+XONE+.xCNOT
CONT2:X1P←X1PRIM
 X2P←X2PRIM
 X←XVECT
 'TO GET A FULL LISTING OF THE RESULTS TYPE:'
 ''
 '                OUTPUT'
 ''
 'TO GET A SHORT LISTING OF THE RESULTS TYPE:'
 ''
 '                SHORT'
 TIME← 60 60 60 ⊤(⍳21)-TIME
 ''
 ''
∇
```

Once the set of matrices $\{C_0, A_n, B_n, n=0,\ldots,k\}$ have been initialized, the function RUN is executed. This function calls all of the main functions in the program and computes all of the desired quantities of interest. If the value of ρ is greater than or equal to .99, the execution is halted. This function also utilizes all of the accuracy checks proposed in this paper.

```
     ∇  X←XVECT;A1INV;AF;BF;BVECT;J;K;K1;L;F;X1;Y;Y1;Y2;T1;
        ITS;Q;R
[1]      X←XNOT,XONE+0×J←1
[2]      K1←⌈X1E+3×SIGMA←(X2E+X1E-X1E*2)*0.5
[3]      A1INV←⌹I-A[1;;]
[4]      BVECT←XNOT+K←0
[5]      BF←(⍴B)⍴AF←(⍴A)⍴R←0×Q←1
[6]      AF[0;;]←A[0;;]+.×A1INV
[7]     LOOP1:AF[Q;;]←A[Q←Q+1;;]+.×A1INV
[8]      →((1↑⍴A)>Q+1)/LOOP1
[9]     LOOP2:BF[R;;]←B[R←R+1;;]+.×A1INV
[10]     →((1↑⍴B)>R+1)/LOOP2
[11]    LOOP3:→(K<1↑⍴BF)/ONE
[12]     BF←BF,[0]0
[13]     AF←AF,[0]0
[14]    ONE:BVECT←BVECT,(XONE+.×AF[K←K+1;;])+XNOT+.×BF[K;;]
[15]     →((K≤2)∨K<K1)/LOOP3
[16]     X←X,((K÷J)×M)⍴ITS←0
[17]    START:ITS←ITS+L÷L←1+J←F←1
[18]     X1←X+1↑Y←Y1←Y2←M⍴0
[19]    SPECIAL:T1←Y+X[(N+L×M)+⍳M]+.×AF[0;;]
[20]     X[(N+(L-1)×M)+⍳M]←BVECT[(N+(L-1)×M)+⍳M]+T1
[21]     →((K+0×J←F←F+1)<L←L+(⍳0)⍴1+1↑Y←Y1←Y2)/TEST
[22]    LOOP4:Y←Y+Y1←X[(N+(L-J)×M)+⍳M]+.×AF[J;;]
[23]     →(2≤J←J-1)/LOOP4
[24]     →SPECIAL
[25]    TEST:→(MAXIT×<ITS)/OUT
[26]     →(1.000000E¯8 <⌈/|X1-X)/START
[27]     →(0.0001<1-+/X+0×J←K+1)/LOOP3
[28]     X←(-M)↓X
[29]     →0
[30]    OUT:'TOO MANT ITERATIONS FOR THE VECTOR X'
[31]     X←(-M)↓X
     ∇
```

The function XVECT computes the stationary vector $\underline{x}$ defined in Equation (3) by using the iterative procedure proposed in Section 7. We first compute the index K1 which is equal to the smallest integer greater than $\mu+3\sigma$ where $\mu = \underline{X}'(1)\underline{e}$, and $\sigma = \sqrt{\underline{X}^{(2)}(1)\underline{e}+X'(1)\underline{e}-(\underline{X}'(1)\underline{e})^2}$. The index K1 is used as an initial truncation of the vector $\underline{x}$. We next compute the vector $\underline{b} = (\underline{b}_0,\underline{b}_1,\ldots,\underline{b}_{K1})$ defined in Equation (46). The loop START performs the iterative procedure defined in Equation (46). These iterations are continued until Condition (47) is met or until a specified maximum number of iterations is reached. If Condition (47) is met, we check to see if an adequate number of components have been computed. This is accomplished by computing the difference $1-\sum_{k=0}^{K1} \underline{x}_k\underline{e}$. If this difference is greater than 10^{-8}, we increase K1 by one and repeat the procedure.

```
     ∇  G←SOLVG;A2;A3;B;C1;D;DD;ERR;G1;G2;R
[1]    ⍝THIS FUNCTION WILL SOLVE THE MATRIX
[2]    ⍝FUNCTIONAL EQUATION G=A(G)
[3]     A3←A[DD←D←((B←⍴A)−1)[GITS←0];;]
[4]     G1←0×A1INV←⌹(I←(⍳R)∘.=⍳R←(¯1↑B))−A[1;;]
[5]    LOOP1:G←G1+0×GITS←GITS+D÷D←DD+0×(⍳0)⍴ 1 1 ↑A2←A3
[6]    LOOP2:→(2<D+0× 1 1 ↑A2←A[D←D−1;;]+A2+.×G)/LOOP2
[7]     ERR←⌈/,|G−G1←A1INV+.×A[0;;]+A2+.×G+.×G
[8]     →((C1←MAXITG>GITS)∧(1.000000E¯8 )<ERR)/LOOP1
[9]     →1↑(C1=1)⌽ 10 15
[10]    'AT ';MAXITG;' ITERATIONS, THE BEST APPROXIMATION'
[11]   '    OF G IS:'
[12]    ⎕←G1
[13]   'WITH A MAXIMUM DIFFERENCE BETWEEN ITERATES OF:';ERR
[14]    →0
[15]    G←G1+(⍉(R,R)⍴((1−+/G1)÷+/G2))×G2←G1−G
     ∇
```

The function SOLVG solves the matrix functional equation $G = \sum_{\nu=0}^{\infty} A_\nu G^\nu$ by the iterative procedure given in Equation (12). i.e.

$$G(0) = (I-A_1)^{-1} A_0$$

$$G(n+1) = (I-A_1)^{-1} \left[A_0 + \sum_{\nu=2}^{k} A_\nu G^{\nu}(n) \right] .$$

The quantity in square brackets is evaluated by Horner's algorithm for the computation of a polynomial. As is well-known, this algorithm minimizes both the number of matrix multiplications and the number of matrix summations required. We continue the iteration until a specified maximum number of iterations have been reached or until the condition

$$\max_{1 \le i,j \le m} \left\{ G_{ij}(n+1) - G_{ij}(n) \right\} < 10^{-8},$$

has been reached. Finally a linear extrapolation is applied, which sets the final matrix G equal to $\{G_{ij}\}$, where

$$G_{ij} = G_{ij}(n+1) + \theta_i [G_{ij}(n+1) - G_{ij}(n)], \text{ for } 1 \le i,j \le m.$$

The quantities θ_i are determined so that the row sums of G are equal to one.

```
    ∇  PI←STVECT P;M;PM;M1
[1]    ⍝  THIS FUNCTION WILL CALCULATE THE STATIONARY
[2]    ⍝PROBABILITY VECTOR OF THE STOCHASTIC MATRIX P
[3]     P←((ρ,PM),ρ,PM←(¯1,¯1+1↓M←ρP)↑P)↑P
[4]     PI←PM+.×⌹((MρPM)+(⍳M1)∘.=⍳M1←1↑M←ρP)-P
[5]     PI←,PI,1-+/PI
    ∇
```

The function STVECT calculates the stationary probability vector of an irreducible stochastic matrix P, using using a method suggested by P. Wachter [36]. We write the stationary equations $\underline{\pi}P = \underline{\pi}$, as

$$\sum_{j=1}^{m} j(\delta_{ij}-P_{ji}) = 0, \quad \text{for } i=1,\ldots,m.$$

where δ_{ij} is the Kroneder delta. If we add P_{mi} to both sides of the i-th equation, we have

$$\sum_{j=1}^{m} \pi_j(\delta_{ij}-P_{ji}+P_{mi}) = P_{mi}.$$

Note that the first m-1 equations do not involve the quantity π_m. Wachter has shown that the first m-1 equations,

$$\sum_{j=1}^{m-1} \pi_j(\delta_{ij}-P_{ji}+P_{mi}) = P_{mi},$$

form a nonsingular system.

In matrix notation, we define I to be the identity matrix of order m-1, the vector $\underline{P}_m$ to be the (m-1)-vector whose i-th component is P_{mi}, the matrix P* to be the matrix obtained by deleting the last row and last column of P, and the matrix P_m to be the matrix all of whose rows are identical and equal to $\underline{P}_m$. The above system of equations may be written as

$$\underline{\pi}^*(I-P^*+P_m) = \underline{P}_m,$$

where $\underline{\pi}^* = (\pi_1, \pi_2, \ldots, \pi_{m-1})$.

The function STVECT solves this system for $\underline{\pi}^*$ and then computes π_m by

$$\pi_m = 1 - \sum_{j=1}^{m-1} \pi_j .$$

```
    ∇  MU←MUVECT;Z
[1]    ⍝ THIS FUNCTION CALCULATES THE VECTOR MU
[2]    GTILDA←(⍴G)⍴GVECT←STVECT G
[3]    Z←(DELTAB←DIAG BETA)+.×GTILDA
[4]    MU←+/(GTILDA+I-G)+.×⌹(I-AA)+GTILDA-Z
    ∇
```

The function MUVECT computes the vector

$\underline{\mu} = (I-G+\tilde{G})(I-A+\tilde{G}-\Delta(\underline{\beta})\tilde{G})^{-1}\underline{e}$.

```
    ∇  K←KMATRIX;R;IB
[1]    IB←(⍳R)∘.=⍳R←1↑⍴BNOT
[2]    K←APRIME+(CPRIME←(CNOT+.×⌹IB-BNOT))+.×BPRIME
    ∇

    ∇  L←LMATRIX
[1]    H←(AAPRIME←⌹I-APRIME←ASERIES)+.×CNOT
[2]    L←BNOT+(BPRIME←BSERIES)+.×H
    ∇
```

The functions KMATRIX and LMATRIX compute the matrices K and L respectively, using their defining Equation (19).

```
    ∇  HSTAR←SOLVH;Z;C;T
[1]    Z←(1↓⍴A)⍴0×C←2
[2]   LOOP:→((¯1+1↑⍴A)>C←C+(⍳0)⍴1↑1,,Z←Z+C×A[C+1;;])/LOOP
[3]    Z←Z+A[2;;]
[4]    T←((AA-A[0;;])-APRIME-AAA←Z+.×GTILDA)+.×INVERSMU
[5]    HSTAR←AAPRIME+.×1+T
    ∇
```

```
    ∇  KSTAR←SOLVK;Z;BBB
[1]    BBB←BTILDA∘.×GVECT
[2]    Z←((AA-A[0;;])-APRIME)+AAA+CPRIME+.×((+⌿B)-BPRIME-BBB)
[3]    KSTAR←1+(+/CPRIME)+Z+.×INVERSMU
    ∇
    ∇  DSTAR1←SOLVL;Z;BGTILDA
[1]    BGTILDA←(GMU←GVECT+.×MU)×BTILDA←(+/B1MEAN)-1-+/BNOT
[2]    Z←((+⌿B)+.×INVERSMU←(⌹GTILDA+I-G)+.×MU)+BGTILDA
[3]    DSTAR1←1+(BPRIME+.×(HSTAR←SOLVH)-INVERSMU)+Z
    ∇
```

The functions SOLVH, SOLVK, and SOLVL compute the vector $\underline{h}^*$, $\underline{\kappa}^*$, and $\underline{d}^*$, respectively. These vectors are defined in Equation (22).

```
    ∇  D2←DELTA2
[1]    D2←FI+.×(+/A2←A2MEAN)+(2×A1+.×U1)-RHO×U1
    ∇
    ∇  D3←DELTA3;Z
[1]    Z←(+/A3←A3MEAN)-(U2×RHO)+D2×U1
[2]    D3←FI+.×(3×A1+.×U2)+(3×A2+.×U1)+Z
    ∇
```

The functions DELTA2 and DELTA3 compute the quantities $\delta^{(2)}(1)$ and $\delta^{(3)}(1)$, respectively, using the recurrence relations given in Equation (31).

```
    ∇  U1←U1PRIM
[1]    U1←((A1INV←⌹I-AA-(⍴AA)⍴FI)+.×BETA)-RHO
    ∇

    ∇  U2←U2PRIM;K1;K2
[1]    K1←2×A1INV+.×(A1-RHO×I)+.×U1←U1PRIM
[2]    K2←+/A1INV+.×(A2-(D2←DELTA2)×I)
[3]    U2←K1+K2-2×(V1←V1PRIM)+.×U1
    ∇
```

```
     ∇   U3←U3PRIM;K1;Z
[1]      Z←(-3×D2×U1)+(+/A3)-D3←DELTA3
[2]      K1←A1INV+.×(3×A1+.×U2)+(-3×RHO×U2)+(3×A2+.×U1)+Z
[3]      U3←K1-3×(V1+.×U2)+(V2←V2PRIM)+.×U1
     ∇
```

```
     ∇   V1←V1PRIM
[1]      V1←(PI+.×A1+.×A1INV)-RHO×PI
     ∇
```

```
     ∇   V2←V2PRIM;Z
[1]      Z←2×RHO×V1←V1PRIM
[2]      V2←((PI+.×A2)+(2×V1+.×A1)-(D2×PI)+Z)+.×A1INV
     ∇
```

The above functions compute the successive derivatives of the Perron-Frobenius eigenvectors, $\underline{u}^{(1)}(1)$, $\underline{u}^{(2)}(1)$, $\underline{u}^{(3)}(1)$, $\underline{v}^{(1)}(1)$ and $\underline{v}^{(2)}(1)$, respectively. These derivatives are also defined in the recurrence relations given in Equations (30) and (31).

```
     ∇   X1E←X1MEAN;K1;K2;Z
[1]      A4←(XNOT+.×+⌿B)-XONE+.×A[0;;]
[2]      K1←((1-+/XNOT)×D2)+A4+.×((2×U1)+U2←U2PRIM)
[3]      Z←(+/XNOT+.×B2←B2MEAN)+K1
[4]      K2←(2×XNOT+.×(B1←B1MEAN)+.×(1+U1))+Z
[5]      X1U1P←(X1←(A4+.×A1INV)+(1-+/XNOT)×PI)+.×U1
[6]      X1E←(K2÷2×1-RHO)-X1U1P
     ∇
```

```
     ∇   X1P←X1PRIM;K1
[1]      K1←(X1E←X1MEAN)×PI
[2]      X1P←K1+((A4+XNOT+.×B1)-X1+.×(I-A1))+.×A1INV
     ∇
```

```
     ∇   X2E←X2MEAN;K1;K2;K3;T;Z
[1]      Z←((+⌿B)+.×(K2←(U3←U3PRIM)+3×U2))++/B3←B3MEAN
[2]      K1←XNOT+.×(3×B2+.×(1+U1))+(3×B1+.×(U2+2×U1))+Z
[3]      T←(3×D2×X1+.×U1)+(1-+/XNOT)×D3
[4]      K3←(-XONE+.×A[0;;]+.×K2)+(3×D2×X1E)+T
[5]      X2E←((K1+K3)÷3×1-RHO)-(X1+.×U2)+4×X1P+.×U1÷3
     ∇
```

```
    ∇  X2P←X2PRIM;K1;Z
[1]    K1←(X2E←X2MEAN)×PI
[2]    Z←XNOT+.×P2+2×P1
[3]    X2P←K1+((X1+.×A2)+Z-2×X1P+.×(I-A1))+.×A1INV
    ∇
```

The functions X1PRIM and X2PRIM compute the moment vectors $\underline{X}'(1)$ and $\underline{X}^{(2)}(1)$, respectively. The functions X1MEAN and X2MEAN compute the quantities $\underline{X}'(1)\underline{e}$ and $\underline{X}^{(2)}(1)\underline{e}$, respectively, which are needed in X1PRIM and X2PRIM. All of the above quantities are defined in Theorem 6.

```
    ∇  APRIME←ASERIES;C
[1]    APRIME←A[C←((ρA)-1)[0];;]
[2]   LOOP:→(2<C+0× 1 1 ↑APRIME←A[C←C-1;;]+APRIME+.×G)/LOOP
[3]    APRIME←A[1;;]+APRIME+.×G
    ∇

    ∇  BPRIME←BSERIES;C
[1]    BPRIME←B[C←((ρB)-1)[0];;]
[2]   LOOP:→(1<C+0× 1 1 ↑BPRIME←B[C←C-1;;]+BPRIME+.×G)/LOOP
[3]    BPRIME←B[0;;]+BPRIME+.×G
    ∇
```

The functions ASERIES and BSERIES compute the quantities $\sum_{\nu=1}^{k} A_\nu G^{\nu-1}$ and $\sum_{\nu=1}^{k} B_\nu G^{\nu-1}$, respectively, using Horner's method.

```
    ∇  A1←A1MEAN;C
[1]    A1←(1↓ρA)ρ0×C←1
[2]   LOOP:A1←A1+A[C;;]×C
[3]    →((1↑ρA)>C←C+1)/LOOP
    ∇

    ∇  A2←A2MEAN;C
[1]    A2←(1↓ρA)ρ0×C←2
[2]   LOOP:A2←A2+A[C;;]×C×C-1
[3]    →((1↑ρA)>C←C+1)/LOOP
    ∇
```

```
    ∇  A3←A3MEAN;C
[1]    A3←(1↓⍴A)⍴0×C←3
[2]   LOOP:A3←A3+A[C;;]×C×(C-1)×C-2
[3]    →((1↑⍴A)>C←C+1)/LOOP
    ∇

    ∇  B1←B1MEAN;C
[1]    B1←(1↓⍴B)⍴C←0
[2]   LOOP:B1←B1+B[C;;]×C+1
[3]    →((1↑⍴B)>C←C+1)/LOOP
    ∇

    ∇  B2←B2MEAN;C
[1]    B2←(1↓⍴B)⍴0×C←1
[2]   LOOP:B2←B2+B[C;;]×C×C+1
[3]    →((1↑⍴B)>C←C+1)/LOOP
    ∇

    ∇  B3←B3MEAN;C
[1]    B3←(1↓⍴B)⍴0×C←2
[2]   LOOP:B3←B3+B[C;;]×C×(C+1)×C+2
[3]    →((1↑⍴B)>C←C+1)/LOOP
    ∇
```

The functions A1MEAN, A2MEAN, A3MEAN, B1MEAN, B2MEAN and B3MEAN respectively compute the moment matrices

$$\sum_{\nu=1}^{k} \nu A_\nu, \qquad \sum_{\nu=1}^{k} \nu B_\nu,$$

$$\sum_{\nu=2}^{k} \nu(\nu-1) A_\nu, \qquad \sum_{\nu=2}^{k} \nu(\nu-1) B_\nu,$$

$$\sum_{\nu=3}^{k} \nu(\nu-1)(\nu-2) A_\nu, \qquad \sum_{\nu=3}^{k} \nu(\nu-1)(\nu-2) B_\nu.$$

```
    ∇  DELTAB←DIAG B;N
[1]    ⍝ THIS FUNCTION WILL CREATE A DIAGONAL MATRIX
[2]    ⍝ WITH THE ELEMENTS OF B ALONG THE DIAGONAL
[3]    DELTAB←N⍴B,(N←(⍴B),⍴B)⍴0
    ∇
```

The function DIAG creates the matrix $\Delta(\beta)$ which is a diagonal matrix with the elements of the vector $\underline{\beta}$ along the diagonal.

B. The Global Variables

The following is a list of the global variables.

A - A three dimensional array where A[N;;] = A_n for $n=0,\ldots,k$.

A1 - $\sum_{\nu=1}^{k} \nu A_\nu$

A2 - $\sum_{\nu=2}^{k} \nu(\nu-1)A_\nu$

A3 - $\sum_{\nu=3}^{k} \nu(\nu-1)(\nu-2)A$

A4 - $\underline{x}_0 \sum_{n=1}^{k} B_n - \underline{x}_1 A_0$

A1INV - $(I-A+\Pi)^{-1}$

AA - $A = \sum_{\nu=0}^{k} A_\nu$

AAA - $\sum_{\nu=2}^{k} (\nu-1)A_\nu\tilde{G}$

AAPRIME - $(I - \sum_{\nu=1}^{k} A_\nu G^{\nu-1})^{-1}$

APRIME - $\sum_{\nu=1}^{k} A_\nu G^{\nu-1}$

B - A three dimensional array where B[N;;] = B_{N+1}, N=0,...,k-1.

B1 - $\sum_{\nu=1}^{k} \nu B_\nu$

B2 - $\sum_{\nu=2}^{k} \nu(\nu-1) B_\nu$

B3 - $\sum_{\nu=3}^{k} \nu(\nu-1)(\nu-2) B_\nu$

BBB - $\sum_{\nu=2}^{k} (\nu-1) B_\nu \tilde{G} = \sum_{\nu=1}^{k} \nu B_\nu \underline{e} - (\underline{e}-B_0\underline{e})\underline{g}$

BETA - $\underline{\beta}$

BNOT - B_0

BPRIME - $\sum_{\nu=1}^{k} B_\nu G^{\nu-1}$

CNOT - C_0

CPRIME - $C_0(I-B_0)^{-1}$

D - $\underline{d}$

D2 - $\delta^{(2)}(1)$

D3 - $\delta^{(3)}(1)$

DELTAB - $\Delta(\underline{\beta})$

DSTAR - $\underline{d}^*$

G - G(1)

GITS - The number of iterations required to compute the matrix G.

GMU - $\underline{g}\mu$

GTILDA - $\tilde{G}$

GVECT - $\underline{g}$

H - H(1)

HSTAR - $\underline{h}^*$

I - An m×m identity matrix

INVERSMU - $(I-G+\tilde{G})^{-1}\underline{\mu}$

K - K(1)

KAPPA - $\underline{\kappa}$

KSTAR - $\underline{\kappa}^*$

L - L(1)

M - The order of the A-matrices.

MAXITG - The maximum number of iterations allowed for the computation of the matrix G.

MAXITX - The maximum number of iterations allowed for the computation of the vector $\underline{x}$.

MU - $\underline{\mu}$

N - The order of the matrix B_0.

PI - $\underline{\pi}$

RHO - ρ

SIGMA - $\sigma = \sqrt{\underline{X}''(1)\underline{e} + \underline{X}'(1)\underline{e} - (\underline{X}'(1)\underline{e})^2}$

TIME - The CPU time used in executing the program.

U1 - $\underline{u}^{(1)}(1)$

U2 - $\underline{u}^{(2)}(1)$

U3 - $\underline{u}^{(3)}(1)$

V1 - $\underline{v}^{(1)}(1)$

V2 - $\underline{v}^{(2)}(1)$

X - $\underline{x}$

X1 - $\underline{X}(1)$

X1E - $\underline{X}'(1)\underline{e}$

X1P - $\underline{X}'(1)$

X1U1P - $\underline{X}(1)\underline{u}^{(1)}(1)$

X2E - $\underline{x}^{(2)}(1)\underline{e}$

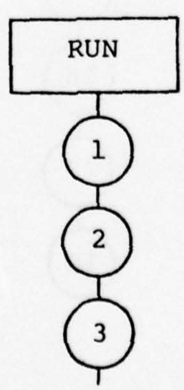
X2P - $\underline{x}^{(2)}(1)$

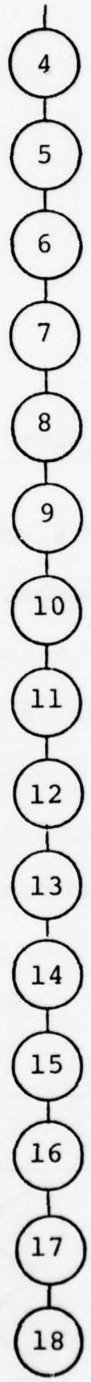
XNOT - $\underline{x}_0$

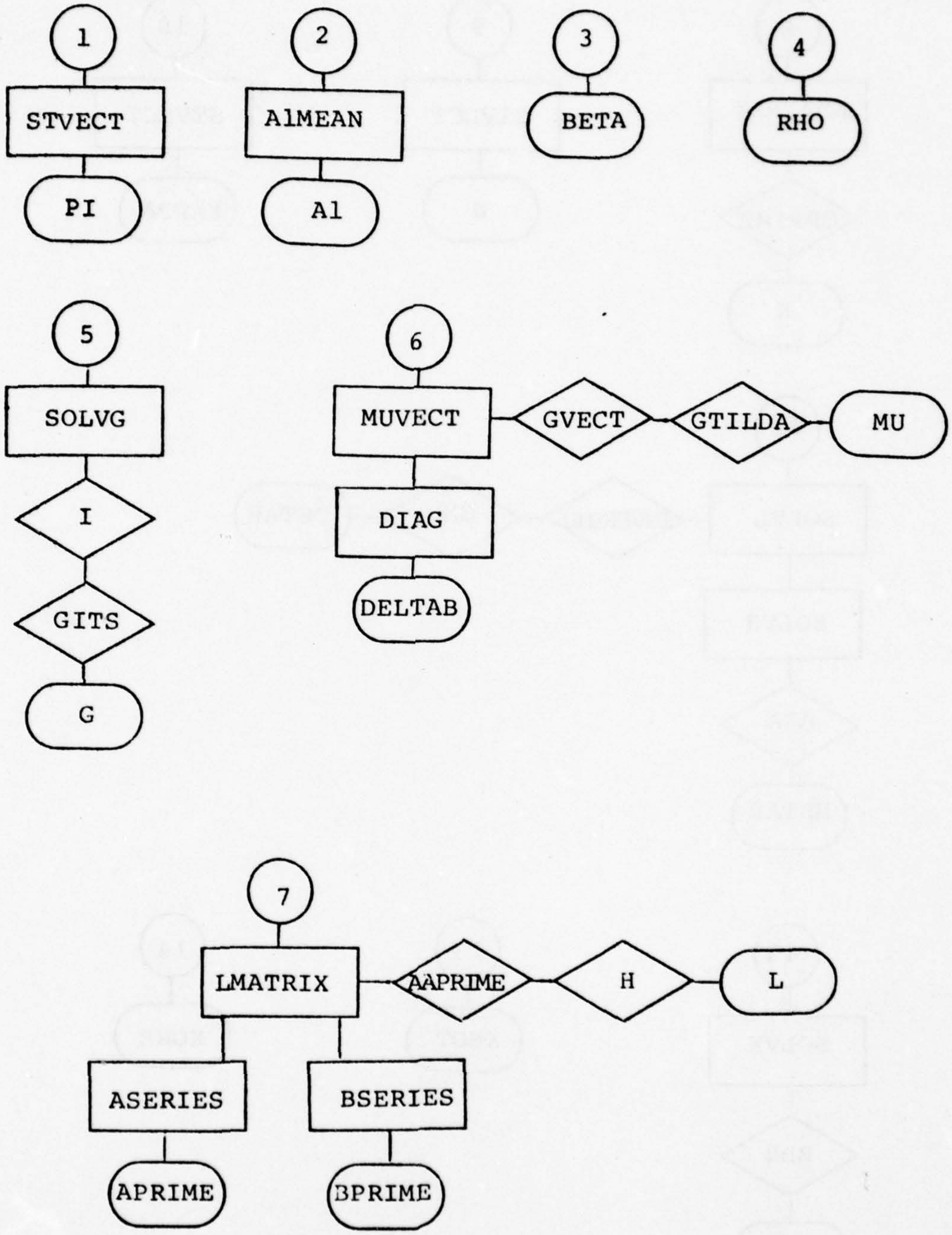
XONE - $\underline{x}_1$

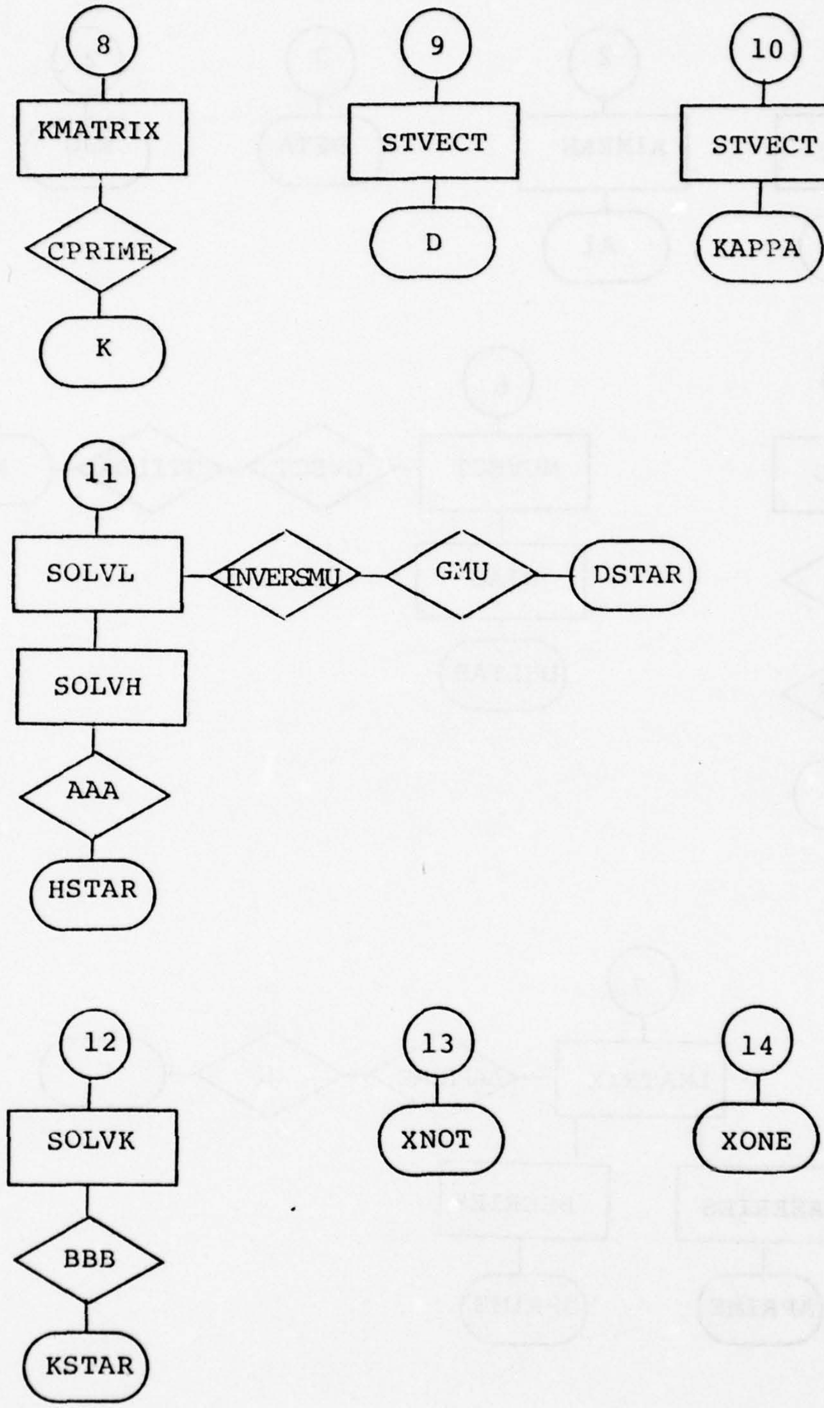
C. The Structure of the APL Program

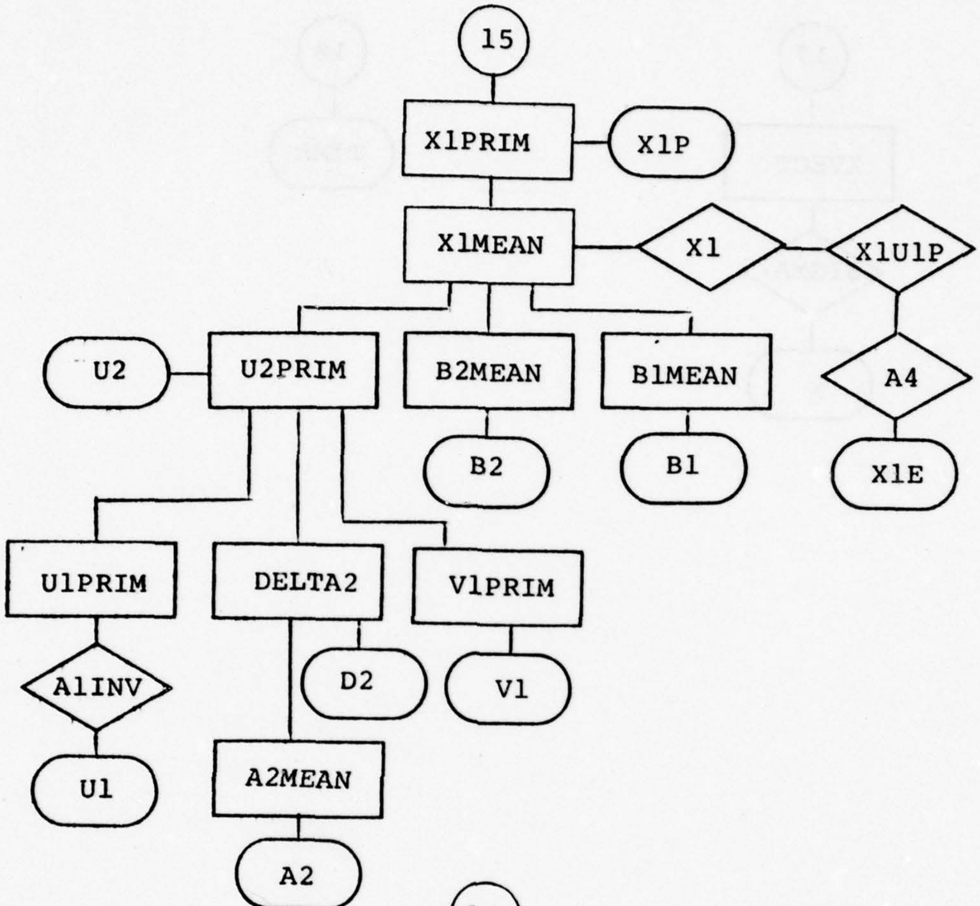
We present below a tree diagram representing the structure of the APL program. We display the order in which the functions are called as well as where each global variable is first initialized. A rectangular box represents the function being called, an oval represents the resultant of that function and a diamond represents any global variables which are initialized internal to the function. We assume that the variables A, AA, B, BNOT, CNOT, M, N, MAXITG, and MAXITX have already been initialized.

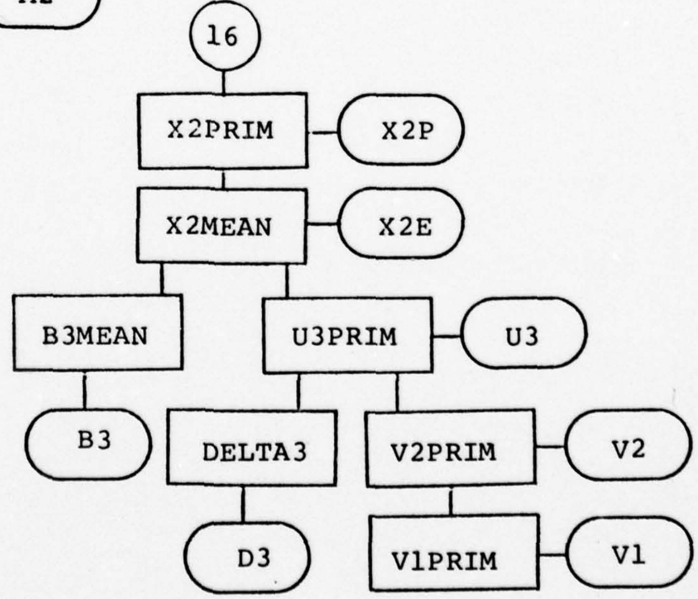


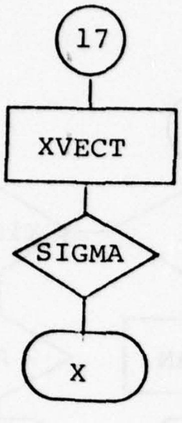

4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

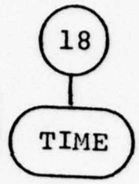

1
STVECT
PI
2
A1MEAN
A1
3
BETA
4
RHO
5
SOLVG
I
GITS
G
6
MUVECT
GVECT
GTILDA
MU
DIAG
DELTAB
7
LMATRIX
AAPRIME
H
L
ASERIES
BSERIES
APRIME
BPRIME


8
KMATRIX
CPRIME
K
9
STVECT
D
10
STVECT
KAPPA
11
SOLVL
INVERSMU
GMU
DSTAR
SOLVH
AAA
HSTAR
12
SOLVK
BBB
KSTAR
13
XNOT
14
XONE


15
X1PRIM
X1P
X1MEAN
X1
X1U1P
U2
U2PRIM
B2MEAN
B1MEAN
A4
B2
B1
X1E
U1PRIM
DELTA2
V1PRIM
A1INV
D2
V1
U1
A2MEAN
A2


16
X2PRIM
X2P
X2MEAN
X2E
B3MEAN
U3PRIM
U3
B3
DELTA3
V2PRIM
V2
D3
V1PRIM
V1


17
XVECT
SIGMA
X


18
TIME

10. SOME NUMERICAL EXAMPLES

We demonstrate the use of the algorithm discussed in this paper by some numerical examples. As we noted in Section 8, in many applications, the detailed structure of the sequences of matrices $\{A_n\}$ and $\{B_n\}$ may be quite involved. In order to avoid cumbersome numerical integrations, we chose to present only simple examples. As such, their practical significance may be somewhat limited, but they do illustrate the concepts discussed in this paper.

Consider a queue with deterministic service times, in which a service time of length c_1 is followed by m-1 services of length c. The next service is again of length c_1, and the pattern is repeated periodically. During a service of length c_1, there are Poisson arrivals of rate λ_1 and during services of length c, there are Poisson arrivals of rate λ. It is easily seen that in this case, $C_0 = A_0$, and that the sequence of matrices $\{B_n\}$ is equal to the sequence of matrices $\{A_n\}$. The m×m matrices A_n are given by

$$A_n = \begin{vmatrix} 0 & e^{-\lambda_1 c_1} \dfrac{(\lambda_1 c_1)^n}{n!} & 0 & 0 & \cdots & 0 \\ 0 & 0 & e^{-\lambda c} \dfrac{(\lambda c)^n}{n!} & 0 & \cdots & 0 \\ 0 & 0 & 0 & e^{-\lambda c} \dfrac{(\lambda c)^n}{n!} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \\ e^{-\lambda c} \dfrac{(\lambda c)^n}{n!} & 0 & 0 & 0 & \cdots & 0 \end{vmatrix}$$

Since $A = \sum_{n=0}^{\infty} A_n$ is doubly stochastic, its invariant probability vector $\underline{\pi}$ is given by $\underline{\pi} = \frac{1}{m} \underline{e}'$. The vector $\underline{\beta}$ is equal to $(\lambda_1 c_1, \lambda c, \ldots, \lambda c)$, and therefore

$$\rho = \underline{\pi\beta} = \frac{1}{m}[\lambda_1 c_1 + (m-1)\lambda c] = \lambda c + \frac{\lambda_1 c_1 - \lambda c}{m} .$$

In the numerical examples, we further simplify by assuming that the arrival rate is constant during all services, i.e. $\lambda_1 = \lambda$, and normalize λ to be equal to one. The matrices A_n are generated by the following program:

```
     ∇  A←GENERATE;BET;N;Z
[1]     LC1←LAMDA1×C1
[2]     LC←LAMDA×C
[3]     BET←LC1,(M−1)⍴LC
[4]     Z←0×A←1⊖DIAG M⍴(*−LC)×N←1
[5]     A[0;1]←(*−LC1)
[6]    LOOP:A←A,[¯1+N≠1]1⊖DIAG M⍴(*−LC)×(LC*N)÷!N
[7]     A[N;0;1]←(*−LC1)×(LC1*N)÷!N
[8]     Z←Z+N×A[N;;]
[9]     →((N←N+1)≤4)/LOOP
[10]    →((⌈/(BET−(+/Z)+N×1−+/+⌿A))>1.000000E¯8 )/LOOP
[11]    A←A,[0](AA←1⊖DIAG M⍴1)−+⌿A
[12]    BNOT←CNOT←A[0;;]
[13]    B← 1 0 0 ↓A
     ∇
```

Assuming the parameters LAMDA, LAMDA1, C, and C1 have been initialized, the matrices A_n are generated until the maximum element in the vector $\underline{\beta} - \sum_{\nu=0}^{n} \nu A_\nu \underline{e} - (n+1)\left[\underline{e} - \sum_{\nu=0}^{n} A_\nu \underline{e}\right]$ is less than 10^{-8}. That this is an adequate truncation has been shown in Neuts [24]. The last matrix is added to the sequence to ensure that the computed matrix A is stochastic. We ran the program several times with m=5, varying the parameters C and C1.

The results of the first run are shown with a complete listing of the output and the remaining runs are shown with an abbreviated listing. Also, to conserve space, we only print 5 digits although APL computes to 18 digits and all of the accuracy checks hold to 13 digits at least.

COMPUTATION OF THE STATIONARY DISTRIBUTION OF
THE QUEUE LENGTH

****

M EQUALS 5

THE ARRIVAL RATE LAMDA = 1

THE ARRIVAL RATE LAMDA1 = 1

THE SERVICE TIME C = 0.1

THE SERVICE TIME C1 = 1.6

THE MATRIX A IS;

```
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
1 0 0 0 0
```

THE ROW SUMS OF A ARE;

1 1 1 1 1

****

COMPUTATION OF THE AUXILLARY QUANTITIES

****

THE VECTOR PI IS;

0.2 0.2 0.2 0.2 0.2

THE VECTOR BETA IS;

1.6 0.1 0.1 0.1 0.1

RHO EQUALS 0.4

THE MATRIX G IS;

0.078351	0.20972	0.3061	0.25174	0.15409
0.0018222	0.00007411	0.90495	0.081974	0.011177
0.011303	0.00041868	0.00064713	0.9054	0.082233
0.083015	0.0025971	0.0039732	0.0034092	0.90701
0.91275	0.019074	0.028702	0.024258	0.015218

9 ITERATIONS WERE REQUIRED FOR THE COMPUTATION
OF THE MATRIX G

THE VECTOR G IS;

0.28625 0.065771 0.15579 0.22574 0.26645

THE VECTOR MU IS;

2.874 1.1118 1.115 1.1347 1.2797

****
****

AN ACCURACY CHECK ON THE COMPUTATION OF THE
MATRIX G AND THE VECTOR MU AS SHOWN IN COROLLARY
1 IN THE THESIS IS THE FOLLOWING;

THE INNER PRODUCT OF THE VECTORS G AND MU IS;

1.6666666655

THE QUANTITY→ 1÷(1-RHO) IS;

1.6666666655

****
****

THE VECTOR D IS;

0.28625 0.065771 0.15579 0.22574 0.26645

THE VECTOR KAPPA IS;

0.1103 0.297 0.26152 0.19188 0.1393

THE VECTOR DSTAR IS;

2.874 1.1118 1.115 1.1347 1.2797

THE VECTOR HSTAR IS;

2.874 1.1118 1.115 1.1347 1.2797

THE VECTOR KAPPA-STAR IS;

3.0361 5.5253 4.9953 4.4094 3.7609

****

COMPUTATION OF THE INVARIANT VECTORS XNOT AND XONE

****

THE VECTOR XNOT IS;

0.17175 0.039463 0.093473 0.13544 0.15987

THE VECTOR XONE IS;

0.023709 0.06384 0.056213 0.041245 0.029943

****
****

THE ACCURACY CHECK PROPOSED IN COROLLARY 2
VERIFIES THE COMPUTATION OF ALL THE QUANTITIES
INVOLVED IN COMPUTING THE VECTORS XNOT AND XONE.
THE VECTOR XNOT SHOULD BE EQUAL TO THE QUANTITY;

XNOT TIMES BNOT PLUS XONE TIMES CNOT

THE ABOVE QUANTITY IS;

0.17175 0.039463 0.093473 0.13544 0.15987

****
****

COMPUTATION OF THE MOMENTS OF THE STATIONARY
VECTOR X

****

THE CONDITIONAL MEAN QUEUE LENGTHS IN THE
VARIOUS PHASES ARE;

0.17104 1.6298 0.92711 0.49448 0.27168

THE MEAN QUEUE LENGTH IS;

0.69882

THE STANDARD DEVIATION OF THE QUEUE LENGTH IS;

1.2825

****

THE STATIONARY VECTOR X

****

50 COMPONENTS OF THE VECTOR X HAVE BEEN COMPUTED
AND THESE COMPONENTS ADD UP TO 0.99996

THE VECTOR X IN PARTITIONED FORM IS GIVEN BY;

0	1.7175E¯1	3.9463E¯2	9.3473E¯2	1.3544E¯1	1.5987E¯1
1	2.3709E¯2	6.3840E¯2	5.6213E¯2	4.1245E¯2	2.9943E¯2
2	3.4653E¯3	5.1794E¯2	3.0613E¯2	1.5423E¯2	7.2202E¯3
3	8.0459E¯4	2.8137E¯2	1.3235E¯2	5.5538E¯3	2.1587E¯3
4	2.0424E¯4	1.1537E¯2	4.6364E¯3	1.7238E¯3	6.0561E¯4
5	4.8375E¯5	3.8205E¯3	1.3695E¯3	4.6585E¯4	1.5238E¯4
6	9.0949E¯6	1.0689E¯3	3.5236E¯4	1.1219E¯4	3.4791E¯5
7	1.2524E¯6	2.6139E¯4	8.1079E¯5	2.4590E¯5	5.6996E¯6
8	1.2135E¯7	5.7436E¯5	1.7057E¯5	3.1938E¯6	6.1207E¯7
9	7.2033E¯9	1.1606E¯5	1.3537E¯6	2.1333E¯7	3.7920E¯8

```
**********************************************
**********************************************

  THE ACCURACY CHECK PROPOSED IN THE THESIS
FOLLOWING EQUATION (40) IS VERIFIED AS
FOLLOWS;

   THE MAXIMUM DIFFERENCE BETWEEN THE VECTOR
GENERATING FUNCTION X EVALUATED AT ONE AND
THE SUM OF THE PARTITIONED COMPONENTS OF X
(EXCLUDING XNOT) IS;

8.8925E¯6

**********************************************
**********************************************
```

COMPUTATION OF THE STATIONARY DISTRIBUTION OF
THE QUEUE LENGTH

****

M EQUALS 5

THE ARRIVAL RATE LAMDA = 1

THE ARRIVAL RATE LAMDA1 = 1

THE SERVICE TIME C = 0.49

THE SERVICE TIME C1 = 0.04

RHO EQUALS 0.4

THE VECTOR MU IS 1.0754 1.8867 1.8458 1.7708 1.6078

****

THE VECTOR XNOT IS;

0.10467 0.16272 0.11698 0.10937 0.10626

THE VECTOR XONE IS;

0.06469 0.028222 0.061545 0.06409 0.064593

****

THE CONDITIONAL MEAN QUEUE LENGTHS IN THE
VARIOUS PHASES ARE;

0.68219 0.24554 0.54915 0.62403 0.66086

THE MEAN QUEUE LENGTH IS;

0.55235

THE STANDARD DEVIATION OF THE QUEUE LENGTH IS,

0.84426

****

THE VECTOR X IN PARTITIONED FORM IS GIVEN BY;

0	1.0467E¯1	1.6272E¯1	1.1698E¯1	1.0937E¯1	1.0626E¯1
1	6.4690E¯2	2.8222E¯2	6.1545E¯2	6.4090E¯2	6.4593E¯2
2	2.2598E¯2	6.8779E¯3	1.7138E¯2	2.0440E¯2	2.1873E¯2
3	6.1400E¯3	1.6724E¯3	3.5350E¯3	4.8643E¯3	5.6577E¯3
4	1.4751E¯3	3.7869E¯4	6.4993E¯4	9.9668E¯4	1.2741E¯3
5	3.2996E¯4	8.0664E¯5	1.1920E¯4	1.8994E¯4	2.6511E¯4
6	6.9510E¯5	1.5278E¯5	2.1109E¯5	3.4705E¯5	5.2055E¯5
7	1.2841E¯5	5.5046E¯7	3.2949E¯6	5.7887E¯6	9.1236E¯6

****

COMPUTATION OF THE STATIONARY DISTRIBUTION OF
THE QUEUE LENGTH

****

M EQUALS 5

THE ARRIVAL RATE LAMDA = 1

THE ARRIVAL RATE LAMDA1 = 1

THE SERVICE TIME C = 0.4

THE SERVICE TIME C1 = 0.4

RHO EQUALS 0.4

THE VECTOR MU IS 1.6667 1.6667 1.6667 1.6667 1.6667

****

THE VECTOR XNOT IS;

0.12 0.12 0.12 0.12 0.12

THE VECTOR XONE IS;

0.059019 0.059019 0.059019 0.059019 0.059019

****

THE CONDITIONAL MEAN QUEUE LENGTHS IN THE
VARIOUS PHASES ARE;

0.53333 0.53333 0.53333 0.53333 0.53333

THE MEAN QUEUE LENGTH IS;

0.53333

THE STANDARD DEVIATION OF THE QUEUE LENGTH IS,

0.79963

****

THE VECTOR X IN PARTITIONED FORM IS GIVEN BY;

0	1.2000E¯1	1.2000E¯1	1.2000E¯1	1.2000E¯1	1.2000E¯1
1	5.9019E¯2	5.9019E¯2	5.9019E¯2	5.9019E¯2	5.9019E¯2
2	1.6437E¯2	1.6437E¯2	1.6437E¯2	1.6437E¯2	1.6437E¯2
3	3.6252E¯3	3.6252E¯3	3.6252E¯3	3.6252E¯3	3.6252E¯3
4	7.3354E¯4	7.3354E¯4	7.3354E¯4	7.3354E¯4	7.3354E¯4
5	1.4460E¯4	1.4460E¯4	1.4460E¯4	1.4460E¯4	1.4460E¯4
6	2.7718E¯5	2.7718E¯5	2.7718E¯5	2.7718E¯5	2.7718E¯5
7	4.5826E¯6	4.5826E¯6	4.5826E¯6	4.5826E¯6	4.5826E¯6

****

In the first run, C1 is much larger than C. As one would expect, the conditional mean queue length is largest immediately following long services and gradually decreases until a minimum is reached at the beginning of the long service. In the second run, C1 is much smaller than C. Here we note the opposite effect. The queue length immediately following the very short service is smallest, as is to be expected. In the last case, C = C1 and we have an M/D/1 queue with ρ = C. Note that in this case, the mean queue

length agrees with that given by the classical formula

$$\underline{X}'(1)\underline{e} = \rho + \frac{\rho^2}{2(1-\rho)} .$$

The small scale implementations of the algorithm, selected for inclusion here, do not fully illustrate its power to handle the high orders of the matrices likely to arise in practical situations. Examples with m as large as fifteen were generated to test our APL program and runs with m as high as fifty, using a well-written FORTRAN program, are entirely feasible without requiring prohibitively expensive processing times, except in cases where $\rho = \underline{\pi\beta}$ is very close to one. In the latter cases, the practical value of steady-state distributions is in fact questionable.

ACKNOWLEDGEMENT

The authors would like to thank Mr. Ramaswami Vaidyanathan for checking all of the computations involved in this paper. We also thank Ms. Beverly Crowl for the excellent typing of the manuscript.

APPENDIX I

Proof of Theorem 4:

By substituting Equations (24) into (27) we obtain

$$(1')\qquad \underline{d} = \frac{(\underline{dd}^*)}{(\underline{\kappa\kappa}^*)}\,\underline{\kappa}C_0(I-B_0)^{-1}.$$

Since $\frac{(\underline{dd}^*)}{(\underline{\kappa\kappa}^*)}$ is a constant, we must first show that $\underline{\kappa}C_0(I-B_0)^{-1}$ is a left eigenvector of $L=L(1)$. Multiplying L on the left by $\underline{\kappa}C_0(I-B_0)^{-1}$, we have

$$(2')\qquad \underline{\kappa}C_0(I-B_0)^{-1}L = \underline{\kappa}C_0(I-B_0)^{-1}B_0 + \underline{\kappa}C_0(I-B_0)^{-1}\left[\sum_{\nu=1}^{\infty} B_\nu G^{\nu-1}(I-\sum_{\nu=1}^{\infty} A_\nu G^{\nu-1})^{-1}C_0\right].$$

Rearranging the equation $\underline{\kappa}K = \underline{\kappa}$ yields

$$(3')\qquad \underline{\kappa}C_0(I-B_0)^{-1}\sum_{\nu=1}^{\infty} B_\nu G^{\nu-1} = \underline{\kappa}\left[I-\sum_{\nu=1}^{\infty} A_\nu G^{\nu-1}\right].$$

Substitution into (2') yields

$$(4')\qquad \underline{\kappa}C_0(I-B_0)^{-1}L = \underline{\kappa}C_0(I-B_0)^{-1}B_0 + \underline{\kappa}C_0 = \underline{\kappa}C_0(I-B_0)^{-1}.$$

Now since $\underline{de}=1$, it remains to verify that

(5') $\quad \underline{\kappa\kappa}^* = (\underline{dd}^*)\underline{\kappa}C_0(I-B_0)\underline{e}$

(by multiplying equation (1') on the right by $\underline{e}$).

Using Theorem 3 and multiplying the equation for $\underline{\kappa}^*$ on the left by $\underline{\kappa}$, we have

$$(6') \quad \underline{\kappa\kappa}^* = 1 + \underline{\kappa}C_0(I-B_0)^{-1}\underline{e} + \underline{\kappa}\Bigg\{C_0(I-B_0)^{-1} \left[\sum_{\nu=1}^{\infty} B_\nu - \sum_{\nu=1}^{\infty} B_\nu G^{\nu-1} + \sum_{\nu=2}^{\infty}(\nu-1)B_\nu\tilde{G}\right] + \sum_{\nu=1}^{\infty} A_\nu - \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1} + \sum_{\nu=2}^{\infty}(\nu-1)A_\nu\tilde{G}\Bigg\}(I-G+\tilde{G})^{-1}\underline{\mu},$$

and similarly

$$(7') \quad \underline{dd}^* = 1 + \underline{d}\left[\sum_{\nu=1}^{\infty} B_\nu - \sum_{\nu=1}^{\infty} B_\nu G^{\nu-1} + \sum_{\nu=2}^{\infty}(\nu-1)B_\nu\tilde{G}\right](I-G+\tilde{G})^{-1}\underline{\mu} + \underline{d}\sum_{\nu=1}^{\infty} B_\nu G^{\nu-1}\left[I-\sum_{\nu=1}^{\infty} A_\nu G^{\nu-1}\right]^{-1} \Bigg\{\underline{e} + \left[\sum_{\nu=1}^{\infty} A_\nu - \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1} + \sum_{\nu=2}^{\infty}(\nu-1)A_\nu\tilde{G}\right](I-G+G)^{-1}\underline{\mu}\Bigg\}.$$

The equation $\underline{d}L = \underline{d}$ implies

$$(8') \quad \underline{d} = \underline{d}\sum_{\nu=1}^{\infty} B_\nu G^{\nu-1}\left(I-\sum_{\nu=1}^{\infty} A_\nu G^{\nu-1}\right)^{-1}C_0(I-B_0)^{-1},$$

and therefore $\underline{dd}^*$ can be written as

$$(9') \qquad \underline{dd}^* = 1 + \underline{d} \sum_{\nu=1}^{\infty} B_\nu G^{\nu-1}\left[I - \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1}\right]^{-1} \underline{e}$$

$$+ \underline{d} \sum_{\nu=1}^{\infty} B_\nu G^{\nu-1}\left[I - \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1}\right]^{-1} \Bigg\{ C_0 (I-B_0)^{-1}$$

$$\left[\sum_{\nu=1}^{\infty} B_\nu - \sum_{\nu=1}^{\infty} B_\nu G^{\nu-1} + \sum_{\nu=2}^{\infty} (\nu-1) B_\nu \tilde{G} \right] + \sum_{\nu=1}^{\infty} A_\nu$$

$$- \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1} + \sum_{\nu=2}^{\infty} (\nu-1) A_\nu \tilde{G} \Bigg\} (I-G+\tilde{G})^{-1} \underline{\mu}.$$

Comparing (6') with (9'), we see that (5') holds if

$$(10') \qquad \left[\underline{\kappa} C_0 (I-B_0)^{-1} \underline{e}\right] \underline{d} \sum_{\nu=1}^{\infty} B_\nu G^{\nu-1}\left[I - \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1}\right]^{-1} = \underline{\kappa}.$$

By a straight forward substitution, it is easily verified that $\underline{d} \sum_{\nu=1}^{\infty} B_\nu G^{\nu-1}\left[I - \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1}\right]^{-1}$ is a left eigenvector of K.

Rearranging the equations $\underline{d}L=\underline{d}$ and $\underline{\kappa}K=\underline{\kappa}$, yields equation (8') and

$$(11') \qquad \underline{\kappa} = \underline{\kappa} C_0 (I-B_0)^{-1} \sum_{\nu=1}^{\infty} B_\nu G^{\nu-1}\left[I - \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1}\right]^{-1}.$$

Multiplying (11') on the right by $C_0(I-B_0)^{-1}$ and comparing with (8') we see that

(12') $$\underline{d} = \theta\underline{\kappa}C_0(I-B_0)^{-1},$$

for some constant θ. But since $\underline{de} = 1$,

(13') $$\theta = \left[\underline{\kappa}C_0(I-B_0)^{-1}\underline{e}\right]^{-1}.$$

Upon substitution into (12'), multiplying on the right by $\sum_{\nu=1}^{\infty} B_\nu G^{\nu-1}\left[I - \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1}\right]^{-1}\underline{e}$ and using (11'), we finally obtain

(14') $$\left[\underline{\kappa}C_0(I-B_0)^{-1}\underline{e}\right]\underline{d}\sum_{\nu=1}^{\infty} B_\nu G^{\nu-1}\left[I- \sum_{\nu=1}^{\infty} A_\nu G^{\nu-1}\right]^{-1}\underline{e} = \underline{\kappa e} = 1,$$

and the theorem is proved.

APPENDIX II

Proof of Theorem 6:

Differentiating (7) once with respect to z yields

$$(1'')\qquad \underline{X}'(z)[zI-A^*(z)] + \underline{X}(z)[I-A^{*\prime}(z)] = \underline{x}_0 \sum_{k=1}^{\infty} B_k z^k + z\underline{x}_0 \sum_{k=1}^{\infty} kB_k z^{k-1} - \underline{x}_1 A_0.$$

Letting z tend to 1- gives

$$(2'')\qquad \underline{X}'(1)(I-A) + \underline{X}(1)\left[I-\sum_{k=1}^{\infty} kA_k\right] = \underline{x}_0 \sum_{k=1}^{\infty} B_k + \underline{x}_0 \sum_{k=1}^{\infty} kB_k - \underline{x}_1 A_0.$$

We note that I-A is singular but $I-A+\Pi$ is non-singular and $\underline{X}'(1)\Pi = (\underline{X}'(1)\underline{e})\underline{\pi}$. Therefore (2") becomes

$$(3'')\qquad \underline{X}'(1) = \left\{\underline{x}_0 \sum_{k=1}^{\infty} B_k + \underline{x}_0 \sum_{k=1}^{\infty} kB_k - \underline{x}_1 A_0 - \underline{X}(1)\left[I-\sum_{k=1}^{\infty} kA_k\right]\right\} (I-A+\Pi)^{-1} + (\underline{X}_1'(1)\underline{e})\underline{\pi}.$$

Differentiating twice in Equation (7) with respect to z

gives

$$(4'')\qquad \underline{X}''(z)[zI-A^*(z)] + 2\underline{X}'(z)[I-A^{*\prime}(z)] - \underline{X}(z)A^{*\prime\prime}(z)$$

$$= 2\underline{x}_0 \sum_{k=1}^{\infty} kB_k z^{k-1} + z\underline{x}_0 \sum_{k=2}^{\infty} k(k-1)B_k z^{k-1}.$$

As $z \to 1-$ we have

$$(5'')\qquad \underline{X}''(1)[I-A] = \underline{X}(1)A^{*\prime\prime}(1) - 2\underline{X}'(1)[I-A^{*\prime}(1)]$$

$$+ 2\underline{x}_0 \sum_{k=1}^{\infty} kB_k + \underline{x}_0 \sum_{k=2}^{\infty} k(k-1)B_k,$$

or

$$(6'')\qquad \underline{X}''(1) = \left\{\underline{X}(1)\sum_{k=2}^{\infty} k(k-1)A_k - 2\underline{X}'(1)\left[I-\sum_{k=1}^{\infty} kA_k\right]\right.$$

$$\left. + 2\underline{x}_0 \sum_{k=1}^{\infty} kB_k + \underline{x}_0 \sum_{k=2}^{\infty} k(k-1)B_k\right\}(I-A+\Pi)^{-1}$$

$$+ (\underline{X}''(1)\underline{e})\underline{\pi}\ .$$

To determine $\underline{X}'(1)\underline{e}$, we multiply (7) on the right by $\underline{u}(z)$ and differentiate to give

$$(7'')\qquad \underline{X}'(z)\underline{u}(z)[z-\delta(z)] + \underline{X}(z)\underline{u}'(z)[z-\delta(z)]$$

$$+ \underline{X}(z)\underline{u}(z)[1-\delta'(z)]$$

$$= \underline{x}_0\sum_{k=1}^{\infty} B_k z^k \underline{u}(z) + z\underline{x}_0\sum_{k=1}^{\infty} kB_k z^{k-1}\underline{u}(z)$$

$$+ z\underline{x}_0\sum_{k=1}^{\infty} B_k z^k \underline{u}'(z) - \underline{x}_1 A_0 \underline{u}(z) - z\underline{x}_1 A_0 \underline{u}'(z),$$

which implies

$$(8'')\qquad \underline{X}'(z)\underline{u}(z) = \frac{1}{z-\delta(z)}\Bigg\{\underline{x}_0\sum_{k=1}^{\infty} B_k z^k \underline{u}(z) + z\underline{x}_0\sum_{k=1}^{\infty} kB_k z^{k-1}\underline{u}(z)$$

$$+ z\underline{x}_0\sum_{k=1}^{\infty} B_k z^k \underline{u}'(z) - \underline{x}_1 A_0 \underline{u}(z) - z\underline{x}_1 A_0 \underline{u}'(z)$$

$$- \underline{X}(z)\underline{u}(z)[1-\delta'(z)]\Bigg\} - \underline{X}(z)\underline{u}'(z).$$

Letting $z \to 1$ yields

$$(9'')\qquad \underline{X}'(1)\underline{e} = \frac{1}{0}\Bigg\{\underline{x}_0\sum_{k=1} B^k \underline{e} + \underline{x}_0\sum_{k=1}^{\infty} kB_k\underline{e} + \underline{x}_0\sum_{k=1}^{\infty} B_k \underline{u}'(1)$$

$$- \underline{x}_1 A_0 \underline{e} - \underline{x}_1 A_0\ \underline{u}'(1) - \underline{X}(1)\underline{e}(1-\rho)\Bigg\} - \underline{X}(1)\underline{u}'(1).$$

Note that the term in braces equals

$$(10'')\quad \underline{x}_0\sum_{k=1}^{\infty}B_k\underline{e} + \underline{x}_0\sum_{k=1}^{\infty}kB_k\underline{e} + \underline{x}_0\sum_{k=1}^{\infty}.B_k\left[(I-A+\Pi)^{-1}\underline{\beta}-\rho\underline{e}\right]$$

$$-\underline{x}_1A_0\underline{e} - \underline{x}_1A_0\left[(I-A+\Pi)^{-1}\underline{\beta}-\rho\underline{e}\right] - \underline{X}(1)\underline{e}(1-\rho)$$

$$= (1-\rho)\left[\underline{x}_0(\underline{e}-B_0\underline{e})-\underline{x}_1A_0\underline{e}\right] + \underline{x}_0\sum_{k=1}^{\infty}kB_k\underline{e}$$

$$+\left[\underline{x}_0\sum_{k=1}^{\infty}B_k-\underline{x}_1A_0\right](I-A+\Pi)^{-1}\underline{\beta} - \underline{X}(1)\underline{e}(1-\rho)$$

$$= \underline{x}_0\sum_{k=1}^{\infty}kB_k\underline{e} + \left[\underline{X}(1)-(1-\underline{x}_0\underline{e})\underline{\pi}\right]\underline{\beta} - \underline{X}(1)\underline{e} + \underline{X}(1)\underline{e}\rho$$

$$= \underline{x}_0\sum_{k=1}^{\infty}kB_k\underline{e} + \underline{X}(1)\underline{\beta} - \underline{X}(1)\underline{e} = 0,$$

by multiplying (3") on the right by $\underline{e}$ and simplifying. Therefore, we may use L'Hopital's rule on Formula (8") to get

$$(11'')\quad \underline{X}'(z)\underline{u}(z) = \frac{1}{1-\delta'(z)}\Bigg\{2\underline{x}_0\sum_{k=1}^{\infty}kB_kz^{k-1}\underline{u}(z)$$

$$+ 2\underline{x}_0\sum_{k=1}^{\infty}B_kz^k\underline{u}'(z) + z\underline{x}_0\sum_{k=2}^{\infty}k(k-1)B_kz^{k-2}\underline{u}(z)$$

$$+ 2z\underline{x}_0\sum_{k=1}^{\infty}kB_kz^{k-1}\underline{u}'(z) + z\underline{x}_0\sum_{k=1}^{\infty}B_kz^k\underline{u}''(z)$$

$$- 2\underline{x}_1A_0\underline{u}'(z) - z\underline{x}_1A_0\underline{u}''(z) + \underline{X}(z)\underline{u}(z)\delta''(z)\Bigg\}$$

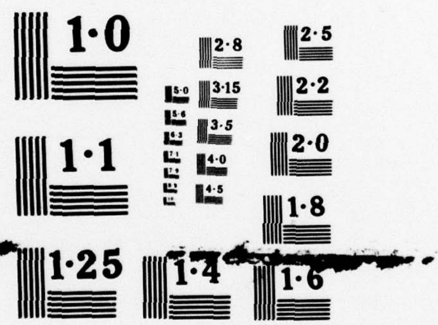
$$- \underline{X}'(z)\underline{u}(z) - 2\underline{X}(z)\underline{u}'(z),$$

which finally yields

AD-A055 946 DELAWARE UNIV NEWARK DEPT OF STATISTICS AND COMPUTER--ETC F/G 12/1
NUMERICAL METHODS FOR A CLASS OF MARKOV CHAINS ARISING IN QUEUE--ETC(U)
MAY 78 D M LUCANTONI, M F NEUTS AFOSR-77-3236
UNCLASSIFIED TR-78/10 AFOSR-TR-78-1049 NL

2 OF 2
ADA
055946

END
DATE
FILMED
8 -78
DDC



NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

$$\text{(12")}\qquad \underline{X}'(1)\underline{e} = \frac{1}{2(1-\rho)}\Bigg\{2\underline{x}_0\sum_{k=1}^{\infty} kB_k\underline{e} + 2\underline{x}_0\sum_{k=1}^{\infty} B_k\underline{u}'(1)$$

$$+ \underline{x}_0\sum_{k=2}^{\infty} k(k-1)B_k\underline{e} + 2\underline{x}_0\sum_{k=1}^{\infty} kB_k\underline{u}'(1)$$

$$+ \underline{x}_0\sum_{k=1}^{\infty} B_k\underline{u}''(1) - 2\underline{x}_1A_0\underline{u}'(1) - \underline{x}_1A_0\underline{u}''(1)$$

$$+ \underline{X}(1)\underline{e}\delta''(1)\Bigg\} - \underline{X}(1)\underline{u}'(1).$$

In order to evaluate $\underline{X}''(1)\underline{e}$, we differentiate (7") with respect to z to get

$$\text{(13")}\qquad \underline{X}''(z)\underline{u}(z)[z-\delta(z)] + 2\underline{X}'(z)u'(z)[z-\delta(z)]$$

$$+ 2\underline{X}'(z)\underline{u}(z)[1-\delta'(z)] + \underline{X}(z)\underline{u}''(z)[z-\delta(z)]$$

$$+ 2\underline{X}(z)\underline{u}'(z)[1-\delta(z)] - \underline{X}(z)\underline{u}(z)\delta''(z)$$

$$= 2\underline{x}_0\sum_{k=1}^{\infty} kB_kz^{k-1}\underline{u}(z) + 2\underline{x}_0\sum_{k=1}^{\infty} B_kz^k\underline{u}'(z)$$

$$+ z\underline{x}_0\sum_{k=2}^{\infty} k(k-1)B_kz^{k-1}\underline{u}(z) + 2z\underline{x}_0\sum_{k=1}^{\infty} kB_kz^{k-1}\underline{u}'(z)$$

$$+ z\underline{x}_0\sum_{k=1}^{\infty} B_kz^k\underline{u}''(z) - 2\underline{x}_1A_0\underline{u}'(z) - z\underline{x}_1A_0\underline{u}''(z),$$

which, upon setting z=1, simplifies to

$$(14'')\qquad \underline{X}''(1)\underline{e} = \frac{1}{0}\Bigg\{\underline{x}(1)\underline{e}\delta''(1) - 2\underline{X}'(1)\underline{e}(1-\rho)$$

$$- 2\underline{X}(1)\underline{u}'(1)(1-\rho) + 2\underline{x}_0\sum_{k=1}^{\infty} kB_k\underline{e}$$

$$+ 2\underline{x}_0\sum_{k=1}^{\infty} B_k\underline{u}'(1) + \underline{x}_0\sum_{k=2}^{\infty} k(k-1)B_k\underline{e}$$

$$+ 2\underline{x}_0\sum_{k=1}^{\infty} kB_k\underline{u}'(1) + \underline{x}_0\sum_{k=1}^{\infty} B_k\underline{u}''(1) - 2\underline{x}_1A_0\underline{u}'(1)$$

$$- \underline{x}_1A_0\underline{u}''(1)\Bigg\} - 2\underline{X}'(1)\underline{u}'(1) - \underline{X}(1)\underline{u}''(1).$$

In order to show that the term in braces equals zero, note that from (10"),

$$(15'')\qquad \underline{x}_0\sum_{k=1}^{\infty} kB_k\underline{e} + \underline{x}_0\sum_{k=1}^{\infty} B_k\underline{u}'(1) = \underline{x}_1A_0\underline{e} + \underline{x}_1A_0\underline{u}\,'(1)$$

$$+ \underline{X}(1)\underline{e}(1-\rho) - \underline{x}_0\sum_{k=1}^{\infty} B_k\underline{e}.$$

Upon substitution of (10") and (15") into the braces and noting that $\underline{x}_1A_0\underline{e} - \underline{x}_0 \sum_{k=1}^{\infty} B_k\underline{e} = \underline{x}_1A_0\underline{e} - \underline{x}_0(\underline{e}-B_0\underline{e}) = 0$, we see that the term in braces reduces to 0. Therefore, we can again apply L'Hopital's rule to get

$$(16'')\qquad \underline{X}''(z)\underline{u}(z) = \frac{1}{3(1-\delta'(z))}\Bigg\{3\underline{X}'(z)\underline{u}(z)\delta''(z)$$

$$+\ 3\underline{X}(z)\underline{u}'(z)\delta''(z) + \underline{X}(z)\underline{u}(z)\delta'''(z)$$

$$+\ 3\underline{x}_0\sum_{k=2}^{\infty} k(k-1)B_k z^k \underline{u}(z) + 6\underline{x}_0\sum_{k=1}^{\infty} kB_k z^{k-1}\underline{u}'(z)$$

$$+\ 3\underline{x}_0\sum_{k=1}^{\infty} B_k z^k \underline{u}''(z) + z\underline{x}_0\sum_{k=3}^{\infty} k(k-1)(k-2)B_k z^{k-3}\underline{u}(z)$$

$$+\ 3z\underline{x}_0\sum_{k=2}^{\infty} k(k-1)B_k z^{k-2}\underline{u}'(z) + 3z\underline{x}_0\sum_{k=1}^{\infty} kB_k z^{k-1}\underline{u}''(z)$$

$$+\ z\underline{x}_0\sum_{k=1}^{\infty} B_k z^k u'''(z) - 3\underline{x}_1 A_0\underline{u}''(z)$$

$$-\ z\underline{x}_1 A_0\underline{u}'''(z)\Bigg\} - \frac{4}{3}\,\underline{X}'(z)\underline{u}'(z) - \underline{X}(z)\underline{u}''(z).$$

Letting $z \to 1-$ yields Equation (44) and completes the proof.

The higher factorial moments of the queue length may in principle be computed in the same manner but the formulas become uninspiringly complicated and will not be shown here.

BIBLIOGRAPHY

[1] Bailey, N.T.J. (1954)
On Queueing Processes with Bulk Service.
J. Royal Stat. Soc., B16, 80-87.

[2] Çinlar, E. (1967)
Time Dependence of Queues with semi-Markovian Services.
J. Appl. Prob., 4, 356-364.

[3] Çinlar, E. (1967)
Queues with semi-Markovian Arrivals.
J. Appl. Prob., 4, 365-379.

[4] Çinlar, E. (1969)
Markov Renewal Theory.
Adv. Appl. Prob., 1, 123-187.

[5] Deb, R.J. and Serfozo, R.F. (1972)
Optimal Control of Batch Service Queues.
Tech. Report, Dept. of I.E. and O.R., Syracuse Univ., Syracuse, N.Y.

[6] Ezhov, I.I. and Shakhbazov, A.A. (1975)
On a Class of Branching Processes.
Prob. Th. and Appl., 20, 180-185.

[7] Gaver, D. (1963)
A Comparison of Queue Disciplines when Service Orientation Times Occur.
Nav. Res. Logist. Quart., 10, 219-235.

[8] Gross, D. and Harris, C.M. (1974)
Fundamentals of Queueing Theory
John Wiley & Sons, New York, N.Y.

[9] Hunter, J.J. (1969)
On the Moments of Markov Renewal Processes.
Adv. Appl. Prob., 1, 188-210.

[10] Karlin, S. and Taylor, H. (1975)
A First Course in Stochastic Processes
Academic Press Inc., New York, N.Y.

[11] Kemeny, J. and Snell, J.L. (1960)
Finite Markov Chains
Van Nostrand, Princeton, N.J.

[12] Loynes, R.M. (1962)
A Continuous-time Treatment of Certain Queues and Infinite Dams.
J. Austral. Math. Soc., 2, 484-498.

[13] Moran, P.A.P. (1954)
A Probability Theory of Dams and Storage Systems.
Austral. J. Appl., Sc., 5, 116-124.

[14] Nair, S.S. and Neuts, M.F. (1972)
Distribution of Occupation Time and Virtual Waiting Time of a General Class of Bulk Queues.
Sankhya, A 34, 17-22.

[15] Neuts, M.F.
Two Markov Chains Arising from Examples of Queues with State-Dependent Service Times.
Sankhya, A 29, 259-264, 1967.

[16] Neuts, M.F. (1966)
The Single Server Queue with Poisson Input and semi-Markov Service Times.
J. Appl. Prob., 3, 202-230.

[17] Neuts, M.F. (1967)
A General Class of Bulk Queues with Poisson Input.
Am. Math. Stat., 38, 759-770.

[18] Neuts, M.F. (1968)
Two Queues in Series with a Finite Intermediate Waitingroom.
J. Appl. Prob., 5, 123-142.

[19] Neuts, M.F. (1970)
Two Servers in Series, Treated in Terms of a Markov Renewal Branching Process.
Adv. Appl. Prob., 2, 110-149.

[20] Neuts, M.F. (1971)
A Queue Subject to Extraneous Phase Changes.
Adv. Appl. Prob., 3, 78-119.

[21] Neuts, M.F. (1973)
The Markov Renewal Branching Process.
Proc. Conf. Math. Meth. in Th. of Queues, Kalamazoo, MI, Springer Verlag, New York, 1974, 1-21.

[22] Neuts, M.F. (1975)
Probability Distributions of Phase Type
--in Liber Amicorum Professor Emeritus H. Florin--
Department of Mathematics, University of Louvain, Belgium, 173-206.

[23] Neuts, M.F. (1976)
Renewal Processes of Phase Type.
Tech. Report #8/76, Dept. of Stat. & C.S., University of Delaware, Newark, DE 19711; forthcoming in Nav. Res. Log. Quart.

[24] Neuts, M.F. (1976)
Moment Formulas for the Markov Renewal Branching Process.
Adv. Appl. Prob., 8, 690-711.

[25] Neuts, M.F. (1976)
Queues Solvable without Rouche's Theorem.
Technical Report No. 76/2, Department of Statistics and Computer Science, University of Delaware, Forthcoming in Opns. Res.

[26] Neuts, M.F. (1976)
Markov Chains with Applications in Queueing Theory, which have a Matrix-geometric Invariant Probability Vector
Tech. Report #76/7, Dept. of Stat. & C.S., University of Delaware, Newark, DE 19711.

[27] Neuts, M.F. (1977)
The M/G/1 Queue with Several Types of Customers and Change-over Times.
Adv. Appl. Prob., 9, 604-644.

[28] Neuts, M.F. (1977)
The M/M/1 Queue with Randomly Varying Arrival and Service Rates.
Tech. Report #77/11, Dept. of Stat. & C.S., University of Delaware, Newark, DE 19711.

[29] Neuts, M.F. (1977)
Some Explicit Formulas for the Steady-state Behavior of the Queue with Semi-Markovian Service Times.
Adv. Appl. Prob., 9, 141-157.

[30] Neuts, M.F. (1977)
A Versatile Markovian Point Process.
Tech. Report #77/13, Dept. of Stat. & C.S., University of Delaware, Newark, DE 19711.

[31] Neuts, M.F. (1978)
Further Results on the M/M/1 Queue with Randomly Varying Rates.
Tech. Report #78/4, Dept. of Stat. and C.S., University of Delaware, Newark, DE 19711.

[32] Purdue, P. (1974)
The M/M/1 Queue in a Markovian Environment.
Opns. Res., 22, 562-569.

[33] Purdue, P. (1975)
A Queue with Poisson Input and semi-Markov Service Times: Busy Period Analysis.
J. Appl. Prob., 12, 353-357.

[34] Ramaswami, V. (1978)
The N/G/1 Queue and its Detailed Analysis.
Ph.D. Thesis, Purdue University.
Tech. Report #78/1, Dept. of Stat. and C.S., University of Delaware, Newark, DE 19711.

[35] Suzuki, T. (1964)
On a Tandem Queue with Blocking.
Journ. Oper. Res. Soc. Japan, 6, 137-157.

[36] Wachter, P. (1973)
Solving Certain Systems of Homogeneous Equations with Special Reference to Markov Chains.
M. Sc. Thesis, Dept. of Math., McGill Univ., Montreal, Canada, 77 pp. with Appendices.

[37] Wallace, V.L. (1969)
The Solution of Quasi Birth and Death Processes arising from Multiple Access Computer Systems.
Ph.D. Thesis, Systems Engineering Lab., Univ. of Michigan, Ann Arbor Mi.
Tech. Report # 07742-6-T.

[38] Yechiali, U. (1971)
Queueing Problems with Heterogeneous Arrivals and Service.
Opns. Res., 19. 722-734.

[39] Yechiali, U. (1973)
A Queueing-Type Birth-and-Death Process defined as a Continuous-time Markov Chain.
Opns. Res., 21, 604-609.